
DISEÑO Y DESARROLLO ELECTRÓNICO DEL CONTROL AUTOMÁTICO DEL LLENADO Y VACIADO DE UNA MÁQUINA DE COMPRESIÓN RÁPIDA



Escuela Técnica Superior de Ingeniería del Diseño

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL
Y AUTOMÁTICA

Autor: Julio Castillo Ruiz

Tutor: José Javier López Sánchez

Curso Académico: 2016-2017

Trabajo de fin de grado

Resumen

En la actualidad, los carburantes llevan un gran control debido a las emisiones de CO₂ que se desprenden al medio ambiente. Para reducir esta contaminación se hacen estudios con diferentes combinaciones químicas en las que luego se examinan los resultados para determinar que carburante es el que cumple con los requisitos.

Estos estudios sirven también para conseguir carburantes más eficientes en los cuales se consigue una mayor potencia y un menor consumo.

Para realizar estos estudios, se necesitan equipos de investigación que facilitan la simulación de la combustión de un motor. Uno de estos equipos, es la máquina de compresión rápida (RCM, siglas que provienen del inglés, Rapid Compression Machine).

El objetivo de este proyecto es controlar el proceso de llenado de la (RCM), de forma que se pueda ajustar la cantidad y rapidez en la que se quiere llenar la máquina.

Por otra parte, también se controlará el proceso de limpieza del depósito de mezclas y de la cámara de combustión. De este modo se pueden hacer pruebas sin que se vean alterados los resultados por residuos de los gases de pruebas anteriores.

El programa que controla dichos procesos se ha creado con LABVIEW, que nos permite desarrollar el programa y a la misma vez crear un “scada” en el que el usuario puede modificar los parámetros de ajuste que sean necesarios y visualizar a tiempo real el comportamiento de la máquina.

Como tarjeta de adquisición de datos se ha implementado una placa Arduino Uno. Su función es enviar datos al programa que son recibidos por sus entradas digitales o analógicas y enviar órdenes recibidas por el programa a la máquina por medio de sus salidas digitales o analógicas.

Palabras Clave: Máquina, Arduino, Labview

Resum

Actualment, els carburants porten un gran control a causa de les emissions de CO₂ que es desprenen a l'atmosfera. Per poder reduir aquestes emissions, s'han fet diversos estudis amb diferents combinacions químiques examinant després quin és el carburant que compleix tots els requisits.

Aquests estudis s'han fet servir també per a aconseguir carburants més eficients que gaudeixen d'una major potència i un menor consum.

Per realitzar aquests estudis, es necessiten equips d'investigació que faciliten la simulació de la combustió d'un motor. Un d'aquests equips, és la màquina de compressió ràpida (RCM, sigles que provenen de l'anglès, Rapid Compression Machine).

El principal objectiu d'aquest projecte és controlar el procés d'emplenament de la (RCM), per poder ajustar la quantitat i rapidesa amb la qual es vol omplir la màquina.

Per altra banda, també es controlarà el procés de neteja del depòsit de mescles i de la cambra de combustió. D'aquesta manera es poden fer proves sense que es vegen alterats els resultats pels residus dels gasos produïts a les proves anteriors.

El programa que controla els processos esmentats amb antelació s'ha creat amb LABVIEW, que ens va a permetre desenvolupar el programa i, a més a més crear una "scada" amb la que l'usuari pot modificar els paràmetres d'ajust que siguin necessaris i visualitzar a temps real el comportament de la màquina.

Com targeta d'adquisició de dades s'ha implementat una placa d'Arduino Uno. La seua funció és enviar dades al programa que són rebudes per les seues entrades digitals o analògiques i enviar ordres rebudes pel programa a la màquina per mitjà de les seues sortides digitals o analògiques.

Paraules Clau: màquina, Arduino, Labview

Abstract

Nowadays, great attention has been paid to the CO₂ emissions released to the atmosphere by the process of fuel combustion. In order to decrease the amount of emissions, many studies have been conducted to determine the best chemical mixture to produce fuel.

The purpose of those studies has been also to achieve more efficient fuel in terms of consumption and power.

The right equipment to simulate the engine combustion is essential to perform those studies. The equipment used in the present study was the rapid compression machine (RCM).

On one hand, the aim of the present work is to control the RCM filling process adjusting its amount and speed. On the other hand, the RCM controls the mixture tank and combustion chamber cleaning. The cleaning ensures that the results will not be contaminated by previous combustions.

The software used to control the process has been LABVIEW. The software visual interface allows the user to modify the RCM parameters and real time visualizations of the procedure.

Arduino Uno board has been implemented as data acquisition card. The main functions of this board have been to send data and commands to the program that are digital or analogically received and processed by the machine.

Key words: machine, Arduino, Labview.

Índice

1. Introducción	9
1.1 Motivación, Justificación y objetivos	9
1.1.1 Motivación	9
1.1.2 Justificación	10
1.1.3 Objetivos	10
1.2 Estructura de la memoria	10
2. Fundamentos básicos	12
2.1 Introducción	12
2.2 Arduino	12
2.2.1 Introducción	12
2.2.2 Software	12
2.2.3 Hardware Arduino modelo UNO	14
2.3 LabVIEW	15
2.3.1 Introducción	15
2.3.2 Diagrama de bloques	15
2.3.3 LabVIEW Interface for Arduino (LIFA)	21
3. Descripción de la instalación	23
3.1 Introducción	23
3.2 Máquina de compresión rápida	23
3.3 Bomba de vacío	24
3.4 Caja de conexiones	25
3.5 Electroválvulas	25
3.6 Sensor de presión	26
4. Desarrollo de Hardware y Software	28
4.1 Introducción	28
4.2 Hardware	28
4.2.1 Circuito impreso	28
4.2.1.1 Componentes eléctricos y electrónicos	28
4.2.1.2 Conexiones en circuito impreso	31
4.2.1.3 Montaje del circuito impreso	35
4.2.2 Fuente de alimentación	37
4.2.3 Conexiones entre caja de circuito impreso y caja distribuidora en máquina	38
4.3 Software	39
4.3.1 Introducción	39
4.3.2 Maniobras que debe realizar el software.	40
4.3.3 Estructura de la programación	41
4.3.3.1 Estructura while loop	41
4.3.3.2 Configuración de puerto de comunicación	41
4.3.3.3 Configuración de pines de Arduino	42
4.3.3.4 Indicador de limpieza de depósito	43

4.3.3.5	Indicador de limpieza de máquina	43
4.3.3.6	Indicador de llenado de máquina	43
4.3.3.7	Activar y desactivar salidas de Arduino	44
4.3.3.8	Lectura de presión de depósito	45
4.3.3.9	Lectura de presión de cámara de combustión	46
4.3.3.10	Rutina de limpieza de deposito	47
4.3.3.11	Rutina de limpieza de la cámara de combustión	49
4.3.3.12	Rutina de llenado de la cámara de combustión	52
4.3.4	Estructura del scada	56
4.3.4.1	Bloque de limpieza deposito	57
4.3.4.2	Bloque de limpieza máquina	59
4.3.4.3	Bloque de llenado de máquina	62
4.3.4.4	Bloque de indicadores y forzados manuales de electroválvulas	65
4.3.4.5	Bloque de muestreo de presión en deposito	65
4.3.4.6	Bloque de muestreo de presión en máquina	66
4.3.4.7	Bloque de comunicación de Arduino con Labview	66
4.3.4.8	Botón de manual/automático	67
4.3.4.9	Botón de paro general del sistema	68
5.	Comprobación del sistema	69
5.1	Introducción	69
5.2	Pruebas realizadas	69
5.2.1	Comprobación del cableado de la instalación	69
5.2.2	Comprobación de las tensiones de alimentación	69
5.2.3	Comprobación de la comunicación entre Arduino y Labview	70
5.2.4	Comprobación de las entradas y salidas de Arduino	71
5.2.5	Comprobación de las rutinas de limpieza y llenado	71
6.	Conclusiones y posibles mejoras	72
6.1	Conclusiones	72
6.2	Posibles mejoras	72
7.	Presupuesto	73
7.1	Introducción	73
7.2	Gastos en materiales	73
7.3	Gastos de mano de obra	74
7.4	Gastos totales	74
8.	Bibliografía	75

1. Introducción

1.1 Motivación, Justificación y objetivos

1.1.1 Motivación

En los últimos años se han empleado un gran número de motores de combustión en la industria del automóvil y de la maquinaria industrial. Estas numerosas cifras, han hecho que aumente la preocupación sobre la calidad del aire que está presente en nuestro entorno. Esto se debe a la contaminación que generan las emisiones de estos motores al hacer la combustión.

Debido a este problema, se fijaron una serie de normas estrictas, realizando un proceso de homologación para que se confirme el cumplimiento de dicha normativa. Estas normas fueron implantadas por las agencias medioambientales y por los gobiernos de todo el mundo. En Europa se diseñó y se implantó una serie de normativas en la que cada vez se establecen unos límites más exigentes. Para poder adaptarse a estos tipos de cambios de normativa, las empresas del automóvil han tenido que diseñar nuevas tecnologías para poder cumplir la normativa de las emisiones de CO₂.

También debemos recordar que los recursos naturales como el petróleo, se van agotando por el paso del tiempo. Debido a esta disminución de fuentes de combustible, se crean estudios para poder encontrar otros recursos alternativos.

Para poder realizar este tipo de estudio necesitaremos una máquina de compresión rápida. Con este tipo de máquina, podemos obtener información del comportamiento del combustible cuando es sometido a una serie de pruebas.

Con esta información podemos valorar el comportamiento de las sustancias químicas que se han empleado en las pruebas y poder encontrar el combustible más eficiente y menos contaminante para los motores de combustión.

1.1.2 Justificación

Debido a las pruebas que se realizan en la máquina de compresión rápida con distintos tipos de composiciones químicas, se necesita un sistema de limpieza y llenado del sistema para que no queden residuos de una prueba a otra y se pueda hacer un llenado controlado y preciso. De esta forma, no se verán alterados los resultados obtenidos de la pruebas por dicha máquina.

Para poder realizar el control de limpieza y llenado de forma automática, se desarrollará un sistema en el que mediante programación y una serie de sensores, controlarán unos determinados actuadores, como electroválvulas y bomba de vacío, que se encargarán de realizar el proceso.

1.1.3 Objetivos

Para realizar el trabajo, hay que cumplir una serie de objetivos que se irán cumpliendo durante la evolución de este. Los objetivos son los siguientes:

- Crear un software que sea capaz de realizar la limpieza y el llenado de la cámara de combustión de una forma automática y precisa.
- Desarrollar el hardware para poder gobernar los actuadores y poder recibir señales de los sensores.
- Diseñar un sistema informático en el que queden registrados los valores de las pruebas realizadas.
- Verificación de que el sistema está respondiendo correctamente, repitiéndose los resultados de las pruebas con la misma exactitud.

1.2 Estructura de la memoria

En este apartado se describe la memoria de forma estructurada, donde se explican varios puntos como herramientas utilizadas, diseño y elaboración del sistema de control y comprobación de los resultados de las pruebas realizadas.

- **1 Introducción:** Describe la justificación, la motivación e indica los objetivos propuestos.
- **2 Fundamentos básicos:** Contiene los tipos de software que se han empleado para realizar el trabajo y la explicación de la utilidad de cada uno.
- **3 Explicación de la instalación:** Explica qué tipo de instalación se ha realizado y los componente que han sido utilizados en la misma.

- **4 Software y Hardware desarrollados:** Describe el funcionamiento de la programación que ha sido desarrollada con los distintos softwares y la parte del hardware implementado para poder controlar el sistema.
- **5 Comprobación del sistema:** En este punto se verificará el funcionamiento del sistema, haciendo pruebas y comprobando que el sistema cumple con las expectativas.
- **6 Conclusiones y posibles mejoras:** Se analizará los resultados del sistema y se planteará si es necesario hacer mejoras en el sistema.
- **7 Presupuesto:** En este punto se detallará todos los costes como mano de obra y materiales que han sido necesarios para desarrollar el sistema.
- **8 Bibliografía:** En este apartado se indicará la herramienta teórica de apoyo que ha sido utilizada para el desarrollo del proyecto.

2. Fundamentos básicos

2.1 Introducción

En este punto se explica los conceptos de las herramientas que se han utilizado para poder realizar el desarrollo del sistema. Veremos el entorno de programación de labVIEW, donde se explicará cómo y con qué instrucciones se ha realizado el programa. También veremos el software de Arduino y su hardware hará la función de tarjeta de adquisición de datos. De este modo se podrá tener una idea para poder comprender con más facilidad el desarrollo del sistema diseñado.

2.2 Arduino

2.2.1 Introducción

Arduino es una plataforma de hardware libre, que se basa en un simple circuito impreso donde va situado un microcontrolador que procesa las señales que se reciben y se envían por una serie de pines de entradas y salidas situadas en el mismo circuito, actuando como una tarjeta de adquisición de datos. Por otra parte también tenemos el software que es la herramienta donde se desarrolla el programa de la aplicación que se desea controlar. El software está diseñado para una sencilla programación con lenguaje C++. Este dispositivo tiene la ventaja de tener un tamaño reducido y un bajo coste, estos también son los motivos por los cuales es muy utilizado.

2.2.2 Software

Al software de Arduino se le denomina como IDE, que son las siglas que hacen referencia a entorno de desarrollo integrado. El lenguaje en el que se realiza la programación de Arduino es C++, que hace que el desarrollo del programa sea más asequible.

Este entorno se compone de varias partes: un editor de código, que es el programa que permite realizar el código al usuario. Un compilador, que es el programa que se encarga de identificar posibles fallos de estructuración del código. Un depurador, que es un programa que permite detener el código para que el usuario pueda examinar lo que está ocurriendo y por último, un constructor de interface gráfica (GUI), que es el programa que permite la simplificación del desarrollo de las interfaces graficas del usuario.

Una de las principales ventajas de este entorno, es que se puede instalar en cualquier sistema operativo, haciendo que cualquier usuario pueda instalarlo sin ningún problema en su ordenador.

Para desarrollar el código, lo hacemos mediante la interface del IDE. Este consiste en un entorno que es mostrado en pantalla donde disponemos de una serie de herramientas y librerías para crear el código en la zona de trabajo. Para la transmisión del código creado, se conecta la placa a Arduino al ordenador mediante un puerto USB para transferir el código realizado previamente. En la siguiente figura 1 se muestra la interface en la que el usuario crea el código.



```

/*****
**
**  LVFA_Firmware - Provides Basic Arduino Sketch For Interfacing With LabVIEW.
**
**  Written By:   Sam Kristoff - National Instruments
**  Written On:   November 2010
**  Last Updated: Dec 2011 - Kevin Fort - National Instruments
**
**  This File May Be Modified And Re-Distributed Freely. Original File Content
**  Written By Sam Kristoff And Available At www.ni.com/arduino.
**
*****/

/*****
**
**  Includes.
**
*****/

// Standard includes. These should always be included.
#include <Wire.h>
#include <SPI.h>
#include <Servo.h>
#include "LabVIEWInterface.h"

/*****
**  setup()
**
*****/

```

Figura 1 interface del software

En la figura se puede ver un ejemplo del lenguaje de programación C++ que utiliza el usuario para el desarrollo del código.

2.2.3 Hardware Arduino modelo UNO

Este modelo de Arduino incorpora en su placa varios componentes como el microcontrolador ATmega328, 14 pines que se pueden configurar como entradas y salidas. Con estas entradas y salidas, se puede conectar dispositivos que transmitan y reciban señales de 0 y 5v.

También cuenta con entradas y salidas analógicas. Las entradas analógicas son usadas para recibir señales de sensores en los que varían su voltaje dependiendo su estado. Las salidas analógicas se usan para enviar señales de tipo PWM.

También dispone de un puerto USB para la comunicación y un botón de reset para restablecer otra vez de nuevo el código programado.

En la siguiente figura 2 veremos la estructura de la placa Arduino UNO y cada uno de sus componentes en los cuales nos dará una idea de cómo realizar las conexiones necesarias.



Figura 2 Estructura placa Arduino UNO

En la siguiente figura 3 se muestra todo tipo de información necesaria de las características de la placa Arduino. Con esta información se puede ver las condiciones necesarias para los voltajes y corrientes que pueden soportar los pines de entradas y salidas. También muestra los tipos de memorias que utiliza el microcontrolador, la Velocidad del reloj, el peso y dimensiones de la placa.

Microcontrolador	ATmega328P
Voltaje de operación	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Pines digitales E/S	14 (de los cuales 6 pueden ser salidas PWM)
Pines digitales PWM E/S	6
Pines de entrada analógica	6
CC por pines de E/S	20 mA
CC en Pin 3.3V	50 mA
Memoria Flash	32 KB (ATmega328P) de los cuales 0.5 KB los emplea el código de carga
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz
Longitud	68.6 mm
Ancho	53.4 mm
Peso	25 g

Figura 3 Características placa Arduino UNO

2.3 LabVIEW

2.3.1 Introducción

Labview es un entorno de programación de forma gráfica. El tipo de lenguaje que utiliza para la programación se llama G, que hace referencia a gráfico. A parte de ser un modelo de programar intuitivo, ofrece la representación gráfica en pantalla de elementos virtuales como pulsadores, leds, gráficas y muestreo de datos a tiempo real. El programa está dividido en dos partes, una es el panel frontal que se utiliza para que el usuario interactúe con el programa introduciendo y visualizando datos. La otra parte es el diagrama de bloques, esta se utiliza para el desarrollo del programa donde se usan instrucciones y datos que son interconectados para la funcionalidad del código. A continuación se detallará con más detalle las partes de panel frontal y la parte de diagrama de bloques.

2.3.2 Diagrama de bloques

Esta es la parte donde se crea el programa que posteriormente va a procesar toda la información recibida por el panel frontal o por las entradas de la de la tarjeta de adquisición de datos, en este caso la placa Arduino UNO. Después de

hacer los cálculos necesarios se enviarán los resultados al mismo panel frontal para ser muestreados o a las salidas de la tarjeta de adquisición de datos.

El tipo de lenguaje que se emplea para este tipo de programación es G que hace referencia a gráfico. En la siguiente figura 4 se muestra como es el entorno de trabajo del diagrama de bloques que estamos nombrando.

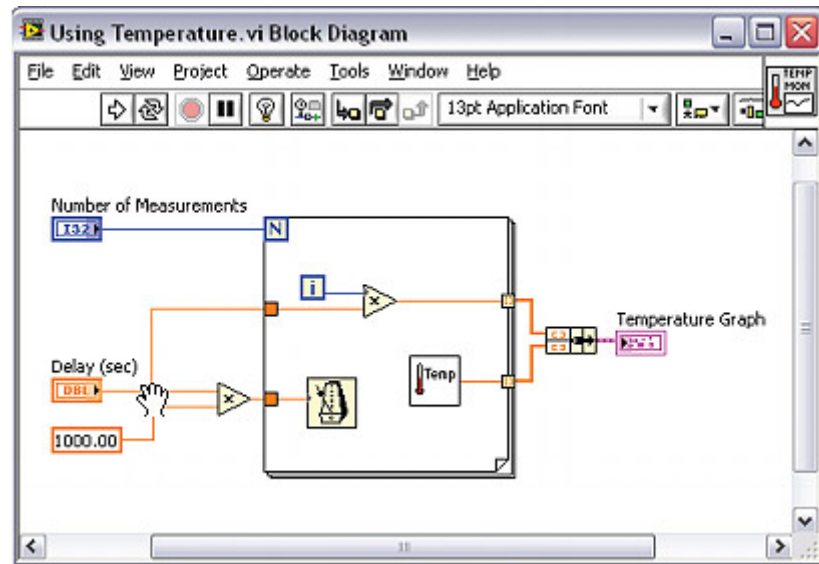


Figura 4 Entorno de trabajo de diagrama de bloques

Como se puede ver en la figura 4, el entorno de trabajo del diagrama de bloques, es muy común a la mayoría de los programas de Windows, donde se encuentra la pestaña de archivo, editar, ver, y otras pestañas comunes a otros entornos de programación. En la zona de trabajo se puede ver una serie de cajas conectadas entre sí mediante unas líneas que hacen de conexión. La caja grande hace referencia a un bucle en el que el programa se va a estar ejecutando durante un tiempo determinado y las cajas pequeñas hacen referencia a instrucciones usadas para resolver de una forma lógica el objetivo del programa realizado.

Para tener un poco más de idea de lo que son y lo que hacen las instrucciones utilizadas para la programación, a continuación se explicará de forma breve la función de cada una de ellas.

En la siguiente figura 5 se puede ver las diferentes clases de estructuras que se utilizan en el diagrama de bloques. A continuación se explicarán solo las que se han utilizado en el TFG.

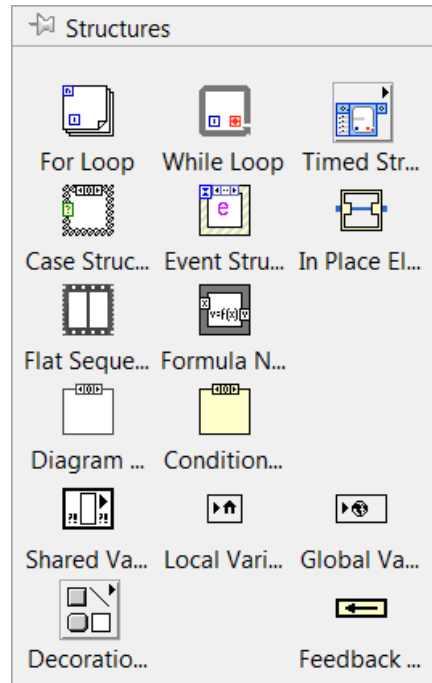


Figura 5 Barra de funciones de structure

- For loop: esta estructura se comporta de la misma forma que el bucle for en programación en C, el bucle se repetirá tantas veces como se haya configurado.
- While Loop: esta estructura actúa de la misma manera que el bucle while de programación en C, el bucle se ejecuta mientras se cumpla la condición dada.
- Case structure: esta estructura puede comportarse de dos formas. Una como si se tratase de un if, else en programación en C, y otra es actuando como una secuencia repitiéndose el bucle de forma ordenada para cada caso.

En la siguiente figura 6 muestra la variedad de clases de instrucciones Numeric que son utilizadas en el diagrama de bloques. En este punto solo se explicarán las que se han utilizado en el TFG.

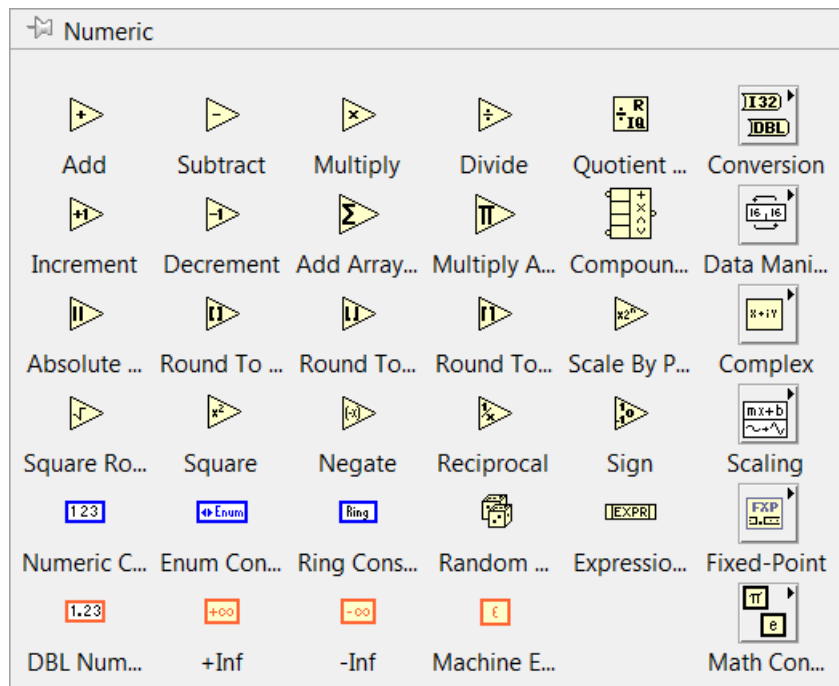


Figura 6 Barra de funciones de Numeric

- Add: se utiliza para realizar la suma de datos numéricos como contantes o valores introducidos por el panel frontal.
- Subtrac: se utiliza para realizar la resta de datos numéricos como contantes o valores introducidos por el panel frontal.
- Multiply: se utiliza para realizar la multiplicación de datos numéricos como contantes o valores introducidos por el panel frontal.
- Divide: se utiliza para realizar la división de datos numéricos como contantes o valores introducidos por el panel frontal.
- Numeric constan: se utiliza para crear una contante en el diagrama de bloques que posteriormente se utilizará para realizar algún cálculo.
- Enum constant: Se crea en el panel frontal y se utiliza para introducir un dato numérico donde hará referencia en el diagrama de bloques para posteriormente ser utilizado para realizar algún cálculo.

En la siguiente figura 7 se puede ver los tipos de clases de instrucciones Boolean que son utilizadas en el diagrama de bloques. A continuación solo se explicarán las que se han utilizado en el TFG.

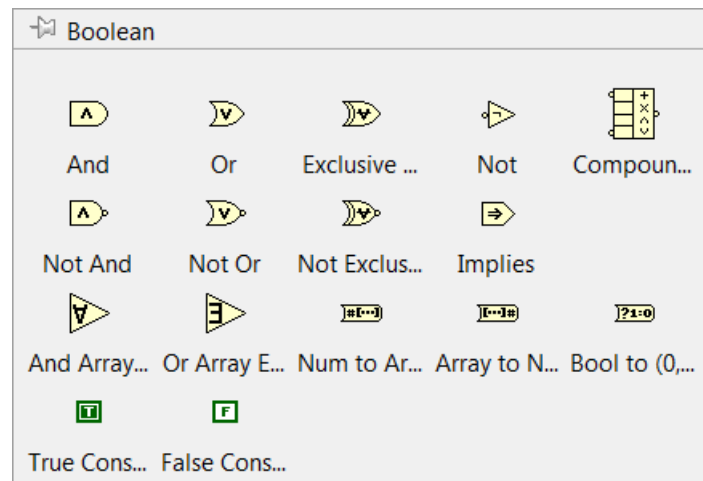


Figura 7 Barra de funciones de Boolean

- And: su función es hacer de multiplicación con datos booleanos, es decir, datos con un valor de verdadero o falso.
- Or: se utiliza para hacer una suma con datos booleanos, es decir, datos con un valor de verdadero o falso.
- Boolean to (0,1): se utiliza para transformar un dato de tipo booleano a un dato de tipo binario 0 o 1.
- True constan: Es una contante de dato booleano y su valor es verdadero. Esta instrucción se utiliza en el diagrama de bloques para operar con más datos de tipo booleano.
- False constan: Es una contante de dato booleano y su valor es Falso. Esta instrucción se utiliza en el diagrama de bloques para operar con más datos de tipo booleano.

En la siguiente figura 8 se puede ver los tipos de clases de instrucciones Timing que son utilizadas en el diagrama de bloques. A continuación solo se explicarán las que se han utilizado en el TFG.

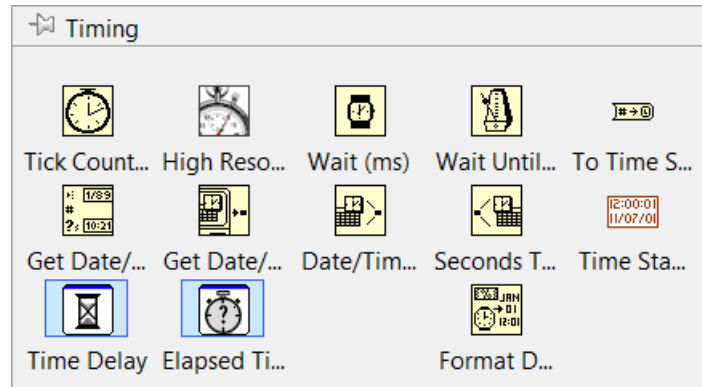


Figura 8 Barra de funciones de Timing

- Wait(ms): esta instrucción se utiliza para el control y tratamiento del tiempo como puede ser la función de un temporizador.

2.3.3 LabVIEW Interface for Arduino (LIFA)

LIFA es un software creado por National instruments que se utiliza como interface para la comunicación de Labview con Arduino. Este software fue creado en 2010 al ver la demanda, las aplicaciones y lo práctica que era la placa Arduino.

Este software se compone de una serie de librerías que hacen que pueda ser posible la comunicación entre Labview y Arduino. Una librería se encarga de instalar en Labview, toda la herramienta necesaria para la programación con instrucciones en las que se gobernarán los pines de entradas y salidas de la placa Arduino. Otra librería se instala en la placa Arduino para que reconozca el puerto de comunicación que establece con Labview y poder actuar como si se tratase de una tarjeta de adquisición de datos.

Una vez instaladas las librerías de LIFA en Labview y la placa Arduino, solo faltará la conexión de estos dos dispositivos mediante un cable de comunicación tipo USB.

En la siguiente figura 9 se muestra una serie de instrucciones que se utilizan en el diagrama de bloques de Labview para poder configurar el puerto, las entradas y las salidas de Arduino. A continuación se explicará de forma breve las instrucciones usadas en el TFG.

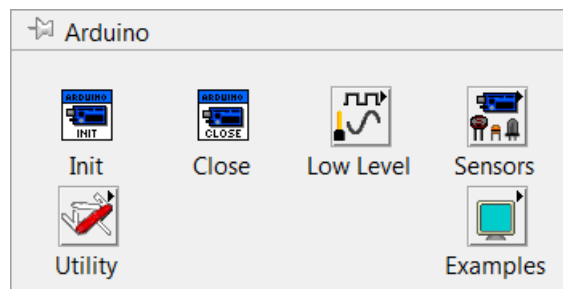


Figura 9 instrucciones generales de LIFA

- Init: esta instrucción se sitúa en el diagrama de bloques de Labview y se utiliza para para inicializar la comunicación de la placa Arduino.
- Close: esta función también se sitúa en el diagrama de bloques y se utiliza para cerrar el lazo de la comunicación entre Labview y Arduino.
- Low level: es un icono de una paleta donde se encuentra la configuración de los pines de Arduino. En la figura 10 se muestra el contenido de las siguientes instrucciones.

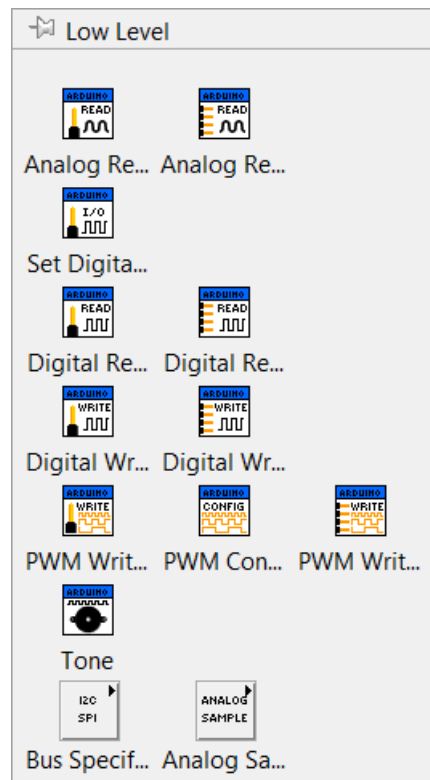


Figura 10 instrucciones Low level

- Analog read: esta instrucción es utilizada para la lectura de una señal analógica de los pines de entrada de la placa Arduino. en la misma instrucción se le configura el número de pin que se quiere usar.
- Set digital: esta instrucción sirve para configurar los pines digitales. En la configuración se le asigna el número de pin que se quiere usar y de qué tipo va a ser, si de entrada o de salida.
- Digital read: esta instrucción se utiliza para leer la señal que está entrando por el pin que ha sido previamente configurada como entrada.
- Digital write: esta instrucción se utiliza para enviar una señal por el pin que ha sido previamente configurado salida.

3. Descripción de la instalación

3.1 Introducción

En este punto se detallan los componentes que van a ser necesarios para el funcionamiento adicional que se va a aplicar en la máquina de compresión rápida (RCM). Previamente se va a explicar cómo es la máquina donde se va a realizar la instalación de la nueva aplicación.

3.2 Máquina de compresión rápida

La máquina de compresión rápida (RCM) es una herramienta muy útil diseñada para la investigación del comportamiento de los combustibles en los motores de combustión. El objetivo que se quiere conseguir con esta máquina es ver la reacción que se produce en la cámara de combustión con las distintas mezclas de combustibles que se introducen en la misma cámara.

La información del comportamiento de las pruebas de la máquina se recoge mediante unos sensores de presión que indican la presión que ha habido en la compresión y la expansión de dicha prueba.

Otra ventaja de esta máquina es que permite realizar una serie de ajustes para poder simular motores de diferentes dimensiones.

En la siguiente figura 11 se muestra la imagen de la máquina de compresión rápida donde se va a realizar la instalación del funcionamiento adicional.

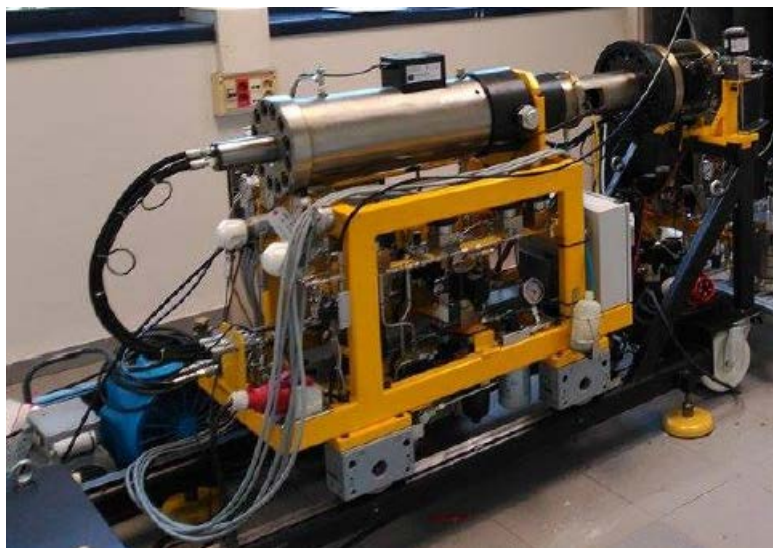


Figura 11 Máquina de compresión rápida

3.3 Bomba de vacío

La bomba de vacío es un dispositivo que se utiliza para la extracción de aire o gas que puede haber en los interiores de depósitos y tuberías de una instalación, llegando a una presión por debajo de la atmosférica. Su funcionamiento es el opuesto al de un compresor, en vez de comprimir, lo que hace es aspirar el gas o aire del circuito. Las características de la bomba de vacío que va a ser utilizada en el proyecto son las siguientes.

Potencia: 27W

Capacidad de vacío: 13 mbar

Alimentación: 230V

Materiales: Cabeza de aluminio y válvulas de Acero inoxidable.

En la siguiente figura 12 se muestra la imagen de la bomba de vacío que se utiliza en la instalación de la máquina.

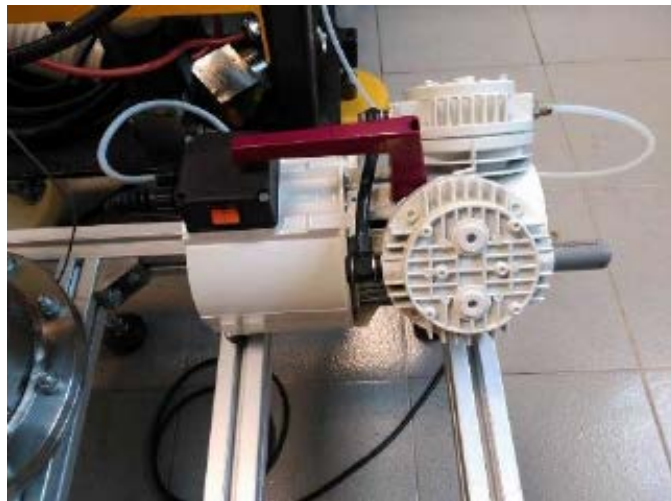


Figura 12 Máquina Bomba de vacío

3.4 Caja de conexiones

En la caja de conexiones se encuentra por una parte el cableado que viene de los relés que gobierna la tarjeta de adquisición de datos, en este caso la placa Arduino. Por otra parte se encuentra el cableado que va a los sensores y actuadores que van distribuidos por la máquina. Todo este cableado es conectado entre sí con su correspondiente cable dentro de la caja de conexiones. En la siguiente figura 13 se puede ver la imagen de la caja de conexiones.



Figura 13 Caja de conexiones

3.5 Electroválvulas

Las electroválvulas que se han instalado en los conductos de circulación de gases de la máquina de compresión rápida, se utilizan para cerrar o abrir el paso de los gases o aire que circula por la instalación. Su función es realizar mediante la excitación de una bobina eléctrica, un movimiento mecánico de una compuerta para abrir o cerrar el paso del componente que circule por la tubería. Las características de dicha electroválvula son las siguientes.

Voltaje de alimentación: 24Vdc

Corriente de alimentación: 750mA

Rango de presiones: 0 a 100 bar

En la siguiente figura 14 se muestra la imagen de una de las electroválvulas que se utilizan en la instalación de la máquina.



Figura 14 Electroválvula

3.6 Sensor de presión

La función del sensor de presión es detectar la presión que se está ejerciendo en un punto o en una instalación de tuberías en circuito cerrado. Los sensores que se han utilizado en el proyecto son sensores de presión eléctricos. Estos sensores ofrecen una pequeña corriente del orden de mA entre los extremos de sus dos cables de conexión, que indican de forma proporcional la presión que se está ejerciendo dependiendo de la variación de la corriente. En la siguiente tabla se puede ver las prestaciones y características que tiene el sensor de presión que se ha utilizado en el proyecto. En este caso el sensor es del fabricante WIKA y el modelo A-10.

Rango de medida	bar	0...10
Límite de sobrepresión	bar	20
Voltaje de entrada (recomendado)	V	8...30
Voltaje de entrada (límite)	V	8...35
Salida	mA	4...20
Precisión	% alcance	$\leq \pm 1$
Rango de temperatura operativo	° C	0...80
Rango de temperatura máximo	° C	-30...100
No repetitividad	% alcance	$\leq 0,1$
Deriva a largo plazo (según IEC 61298-2)	% alcance	$\leq \pm 0,1$
Ruido de señal	% alcance	$\leq \pm 0,3$
Tiempo de establecimiento	ms	< 4 ms

Como se puede observar en la tabla, este sensor tiene un rango de medición de 0 a 10 bar, donde a 0 bar ofrece una corriente de 4 mA y a 10 bar una corriente de 20mA. En la siguiente figura 15 se puede ver la imagen del sensor de presión que se ha utilizado en el proyecto.



Figura 15 Sensor de presión

4. Desarrollo de Hardware y Software

4.1 Introducción

En este punto se va a explicar todo el conjunto desarrollado de hardware y software que se ha implementado en la máquina de compresión rápida, para realizar el llenado y limpieza de forma automática de la cámara de combustión de dicha máquina. Primeramente se explicará la parte de software donde se verán los pasos realizados para el montaje de la instalación eléctrica compuesta por circuito impreso y cableado. Por otra parte se explicará la parte de software donde se detallarán los pasos de la programación que controlará todos los componentes como sensores y actuadores conectados al circuito impreso.

4.2 Hardware

4.2.1 Circuito impreso

Un circuito impreso es una base de baquelita y cobre donde se colocan todos los componentes eléctricos y electrónicos conectados mediante pistas de cobre para realizar una función determinada para alguna aplicación. El diseño y montaje del circuito consta de una serie de pasos. En primer lugar se debe tener bien claro que es lo que va a gobernar el circuito, una vez se tenga clara la idea, se seleccionan los componentes eléctricos y electrónicos necesarios para realizar la función deseada. El siguiente paso es el diseño del circuito, en este paso hay que diseñar la distribución de los componentes sobre el circuito impreso de manera que queden de forma ordenada y que las pistas de cobre puedan llegar al punto de conexión deseado sin intersectar con otras pistas de distintas señales. El último paso es la impresión y montaje del circuito impreso y de los componentes eléctricos y electrónicos.

4.2.1.1 Componentes eléctricos y electrónicos

Los componentes que se van a poner en la placa se van a encargar de suministrar y recibir la corriente necesaria para que los sensores y actuadores instalados en la máquina puedan realizar su función. De esta manera el software diseñado podrá dar instrucciones de las maniobras deseadas de la máquina mediante los componentes circuito impreso. Seguidamente se van a indicar los tipos y cantidades de los componentes instalados en el circuito impreso.

-1 Placa Arduino: esta hará de intermediaria entre el ordenador donde está instalado el software diseñado y los componentes eléctricos y electrónicos para enviar instrucciones a la máquina. Ver placa en la figura 16.



Figura 16 Placa Arduino

-8 Bornes de conexión: los bornes son los puntos donde se van a realizar las conexiones de alimentación, actuadores y sensores. Ver bornes en la figura 17.



Figura 17 Bornes de conexión

-2 Resistencias de 250Ω : estas resistencias serán empleadas para convertir la corriente que ofrece el sensor de presión de 4 a 20mA a una tensión de 0 a 5V para que Arduino pueda procesar la señal de entrada analógica. Para obtener el valor de la resistencia hay que tener en cuenta una serie de datos. En primer lugar se sabe que el máximo voltaje de entrada analógico a la placa Arduino es de 5V y por otro lado se sabe que el valor máximo de corriente que ofrece el sensor de presión es de 0.02 A. si se quiere obtener un valor de 5V en extremos de la resistencia cuando el valor de corriente que ofrece el sensor es de 0.02 A, hay que poner la resistencia que se ha nombrado anteriormente con un valor de 250Ω . En la siguiente formula con la ayuda de la ley de ohm se obtiene el valor de la resistencia que hay que poner para respetar los valores en los que se desea trabajar.

$$R \leq \frac{V}{I} \rightarrow R \leq \frac{5\text{ V}}{0,02\text{ A}} = 250\Omega$$

En la siguiente figura se muestra 18 se muestra la imagen de una resistencia eléctrica.



Figura 18 resistencia óhmica

-6 Relés: estos son los que se encargan de dar paso de la corriente a las electroválvulas instaladas en la máquina para la nueva aplicación. Estos relés se alimentarán a una tensión de 24V y cambiarán de estado sus contactos dependiendo de la alimentación. De estos 6 relés solo se utilizarán 5, el sobrante se dejará de reserva instalado por si se quiere implementar otro actuador más. Ver relé en la figura 19.



Figura 19 relé

-1 Amplificador de señal ULN2003AN: este componente se encarga de amplificar una tensión pequeña de corriente continua a una tensión de 24Vdc. De esta forma se puede alimentar componentes que requiera una tensión de 24vdc sin limitarnos a los 5Vdc que ofrece la placa Arduino a sus salidas. Este dispositivo está formado por una serie de transistores Darlington que realizan la amplificación y unos diodos que hacen de protección de sobre cargas. En la siguiente figura 20 se muestra el amplificador de señal y como están situadas de forma horizontal la entrada correspondiente a la salida de este dispositivo.

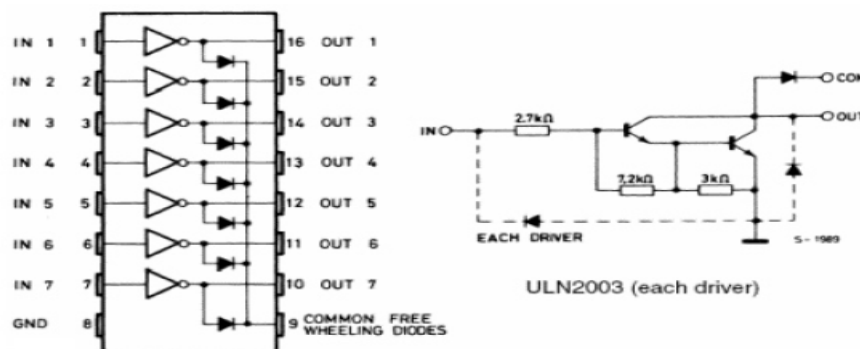


Figura 20 amplificador de señal

-1 Regulador de Tensión MC7812B: este componente se encarga de estabilizar la tensión a 12Vdc, de esta forma ofrece un voltaje de alimentación constante sin variaciones apreciables. Ver regulador de tensión en la figura 21.



Figura 21 regulador de tensión

-2 Condensadores eléctricos: se van a utilizar dos condensadores para eliminar los picos de tensión que puedan ser generados por la fuente de alimentación en la parada y en el arranque. Los valores de los condensadores seleccionados son de 2200 μ F y otro de 470 μ F. Ver condensador eléctrico en la figura 22.



Figura 22 Condensador eléctrico

4.2.1.2 Conexiones en circuito impreso

En este apartado se va explicar los puntos de conexión que se deben realizar para que todos los componentes incorporados en el circuito impreso puedan hacer su función correcta y ejecutar las maniobras diseñadas previamente.

Conexión de estabilizador de tensión: En este punto se recibe por los bornes una tensión de 24Vdc de la fuente de alimentación y se reduce dicho voltaje a 12Vdc de forma estabilizada gracias al estabilizador de tensión para poder entregar la tensión adecuada a la placa Arduino. En la siguiente figura 23 se muestra la conexión del estabilizador de tensión.

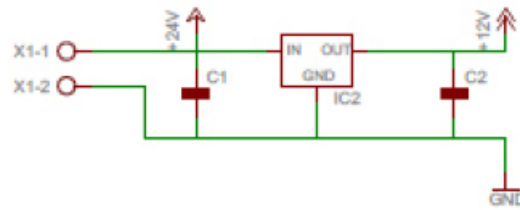


Figura 23 Conexión de estabilizador de tensión

Conexión de los relés: La bobina y el común del contacto de cada uno de los relés que han sido implementados en el circuito impreso, van alimentados a un voltaje de 24Vdc. Una vez se active el relé y se cierre su contacto, este enviará una tensión de 24vdc al actuador que haya conectado en los bornes asignados en el mismo relé. En la siguiente figura 24 se muestra la conexión de lo relés.

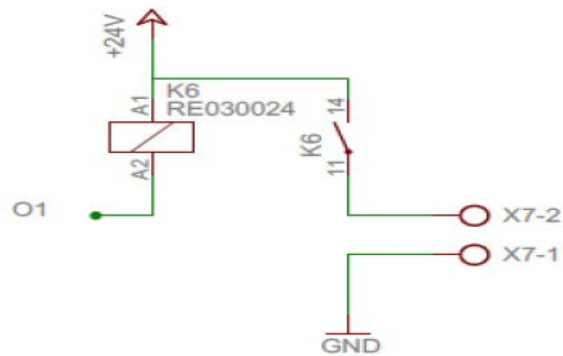


Figura 24 Conexión de relé

Conexión de amplificador de señal: para poder alimentar los actuadores que en este caso son electroválvulas a una tensión de 24Vdc, se necesita poner un amplificador de señal de 5 a 24Vdc ya que la tensión de salida de Arduino viene limitada a 5Vdc. Para cumplir con los requisitos que se necesitan, se ha optado por coger el amplificador ULN2003AN. Las salidas de Arduino se tienen que conectar a las entradas del amplificador y las salidas del amplificador a las electroválvulas. Cada entrada y salida del amplificador está numerada para que se conecte el número de salida con su correspondiente entrada. En la siguiente figura 25 se muestra la conexión del amplificador de señal.

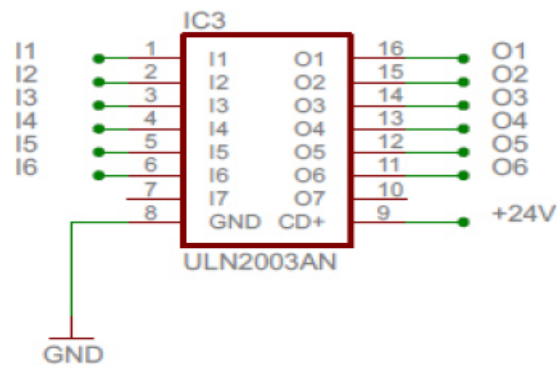


Figura 25 Conexión de amplificador de señal

Conexión de placa Arduino: En la placa Arduino se van a conectar todas las entradas y salidas que van a ser procesadas por el software desarrollado en Labview. Como se va a ver en la imagen siguiente, las entradas analógicas van conectadas en la parte izquierda del pin A0 al pin A5 y las salidas o entradas digitales dependiendo de la configuración que se le haya dado al pin, se conectan en la parte izquierda del pin 0 al 13. Por otra parte también hay que conectar la alimentación de 12Vdc de la placa Arduino entre sus pines +VIN y GND. En la siguiente figura 26 se muestra las conexiones de la placa Arduino.

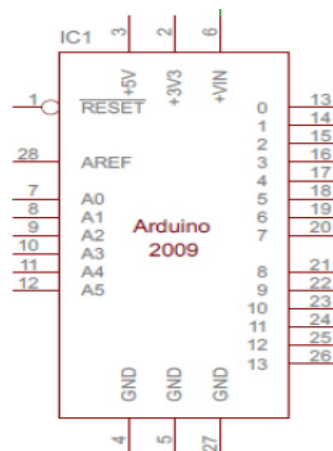


Figura 26 Conexiones en placa Arduino

-Conexión de sensor de presión: los dos cables que nos llegan del sensor de presión se conectan en su bornero específico para él. En una parte del bornero se conectará una tensión de alimentación de 24Vdc y en el otro bornero la señal de corriente que proviene del sensor. La señal de entrada analógica de Arduino va conectada entre el bornero que está conectada la señal del sensor y una resistencia de 250Ω que va conectada a GND. De esta forma en el pin de entrada de analógica de Arduino se recibirá una señal de voltaje en vez de corriente. En la siguiente figura 27 se pueden ver las conexiones del sensor de presión.

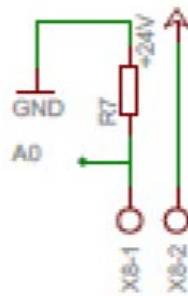


Figura 27 Conexión de sensor de presión

4.2.1.3 Montaje del circuito impreso

Primeramente hay que destacar que este paso lo han realizado los técnicos de electrónica del departamento de motores térmicos, ya que disponen de los medios necesarios y por la complejidad de la tarea que requiere una experiencia mínima.

Para realizar este este paso se ha necesitado una serie de herramientas que se van a nombrar a continuación.

Para el diseño de la distribución de los componentes y las conexiones de los componentes del circuito impreso, se ha utilizado un programa informático específico llamado Eagle que facilita este diseño. El programa consiste en situar los componentes electrónicos sobre una superficie que simula a las dimensiones del circuito impreso. Una vez situados se van realizando las uniones de conexión entre los componentes hasta que quede todo conectado correctamente. Cuando ya está todo conectado, el programa crea las pistas de forma que el circuito quede de una forma ordenada sin que las pistas de distintas señales se intersecten. En la siguiente figura 28 se muestra el resultado de la distribución de componentes y pistas conductoras que ha dado el programa de diseño.

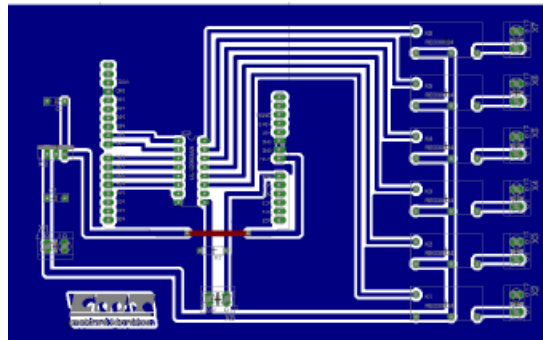


Figura 28 Diseño de circuito impreso

Para transportar el diseño del circuito creado con el programa informático a la placa de baquelita y cobre, se debe de hacer de la siguiente manera. Se imprime el diseño del circuito y se pega ligeramente en la parte de cobre de la placa. Seguidamente, mediante la técnica de grabado fotosensible, se marca el circuito sobre la placa. Una vez marcado, el cobre que sobra se elimina mediante baños en ácido quedando solo las pistas de cobre deseadas.

Una vez se tiene el circuito impreso con las pistas de cobre creadas, se pasa al siguiente paso, que consiste en hacer los agujeros pasantes para introducir las patillas de los componentes. Para este proceso se necesita un taladro, en este caso es un taladro fijo de mesa y una broca de 0,6mm. Una vez se tiene la herramienta necesaria, se apoya el circuito sobre la base del taladro y se taladran los agujeros marcados en el circuito impreso. . En la siguiente figura 29 se muestra el proceso de perforación del circuito impreso.

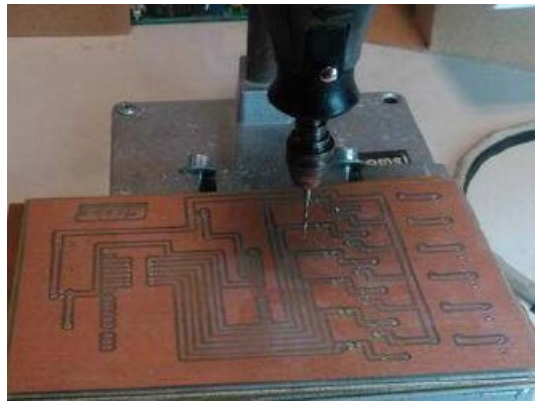


Figura 29 Perforación del circuito impreso

Cuando ya está todo perforado, se pasa al siguiente paso, el paso de soldadura. Para soldar se necesita un soldador de estaño y el estaño necesario para soldar los componentes. La soldadura hay que realizarla por la parte del cobre, de manera que se queden soldadas las patillas de los componentes en las pistas de cobre correspondientes. En la siguiente figura 30 se muestra el proceso de soldadura del circuito impreso.

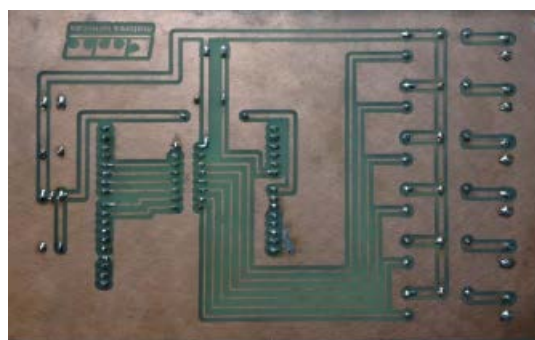


Figura 30 Soldadura del circuito impreso

Una vez ya está el circuito impreso terminado, se pone en la parte de los bornes y componentes para proceder a la conexión del cableado. En el bornero X1 va conectada la alimentación del circuito impreso, del borne X2 la X7 van conectados los actuadores, en esta caso las electroválvulas, y en el borne X8 y X9 van conectados los sensores de presión. En la siguiente figura 31 se muestran las conexiones del circuito impreso.



Figura 31 Conexiones del circuito impreso

4.2.2 Fuente de alimentación

La fuente de alimentación se va a encargar de transformar los 220V de alterna provenientes de la red eléctrica, a una tensión rectificada de 24Vdc que va a ser suministrada al circuito impreso. El modelo de la fuente alimentación que ha sido seleccionada por los técnicos del departamento para soportar toda la corriente demandada Ha sido la LS50-24 de la marca TDK. En la numeración del modelo de la fuente de alimentación indica que es de 50W y de 24Vdc de salida. En la siguiente figura 32 se puede ver la tabla de las características de la fuente de alimentación.

Tensión de salida nominal	24 V DC
Máxima corriente de salida	2.2 A
Máxima potencia de salida	52.8 W
Eficiencia	86 %
Tensión de alimentación	88...264 V AC
Corriente de alimentación	1.3...0.8 A
Rango de Tensión de salida	22...27.2 V DC
Temperatura de operación	-25...70 ° C
Peso	350 g
Dimensiones	99x97x36 mm

Figura 32 Características de la fuente de alimentación

La conexión de la fuente de alimentación es muy sencilla, solo dispone de un bornero con 5 puntos de conexión. En esos 5 puntos van conectados la fase, el neutro y la tierra de los 220Vac, y el negativo más el positivo de los 24Vdc. Los puntos de conexión vienen indicados por el fabricante con una pegatina pegada justamente arriba del mismo bornero. En la siguiente figura 33 se muestra una imagen donde se pueden ver los puntos de conexión de la fuente de alimentación.



Figura 34 Conexiones de la fuente de alimentación

4.2.3 Conexiones entre caja de circuito impreso y caja distribuidora en máquina

Para que las señales de los actuadores y sensores puedan llegar desde del programa diseñado con Lavbview hasta sus puntos de distribución de la máquina, se han diseñado dos cajas de conexión.

Una caja está situada en la misma máquina para distribuir el cableado que va a transportar la señal de cada actuador o sensor. Por una parte salen las mangueras que van a cada actuador o sensor, y por otra parte viene una manguera de la caja del circuito impreso. Esta manguera va conectada a un conector de esta caja. El orden de los pines del conector coincide con el orden de las entradas y salidas de Arduino. En la siguiente figura 35 se muestra una imagen de la caja de conexiones que hay instalada en la máquina.



Figura 35 Caja de conexiones de máquina

La otra caja de conexiones es la que protege la fuente de alimentación y el circuito impreso de posibles golpes y suciedad. En esta caja se realizan las conexiones del cableado proveniente de la caja de conexiones de la máquina y del cableado proveniente de los relés del circuito impreso mediante un conector MIDI. La conexión de este conector esta soldado de forma que el número de pin del mismo, coincide con el número de entrada y salida de la placa Arduino. Por otra parte se realiza la conexión de la placa Arduino con el ordenador por el puerto de comunicación USB. Por este puerto se van a transmitir todas las órdenes del programa diseñado a la placa a la placa Arduino. En la siguiente figura 36 se puede ver la imagen de los conectores que se han nombrado anteriormente.



Figura 36 Caja de conexiones de circuito impreso

4.3 Software

4.3.1 Introducción

En este apartado se van a explicar las maniobras que debe hacer el programa creado con Labview para el llenado y vaciado de depósito y máquina de compresión rápida. El programa gestionará las señales de las entradas y salidas de la tarjeta de adquisición de datos, en este caso la placa Arduino.

El programa está formado por dos partes, una es la que hace de interface con el usuario para ver e introducir datos al sistema. A esta parte se le conoce como Scada en términos técnicos.

Por otro lado está la parte de programación, donde se crean una serie de conexiones entre instrucciones y entradas/salidas para llevar a cabo una maniobra lógica que cubra las necesidades del programador.

4.3.2 Maniobras que debe realizar el software.

El programa tiene que hacer una serie de maniobras que cumpla con las funciones establecidas previamente para el funcionamiento del sistema. Las maniobras que tiene que realizar son las siguientes:

- Todas las salidas que comandan a las válvulas podrán ser forzadas de forma manual e independiente mediante botones virtuales.
- Todas las salidas tienen que tener un piloto virtual independiente que indique en pantalla que dicha salida está siendo activada.
- Se debe mostrar en pantalla las presiones instantáneas de los sensores de presión de depósito y de máquina.
- Tendrá que realizarse un proceso de limpieza de depósito automático cuando el usuario pulse un botón virtual de limpieza de depósito. Cuando se inicie el proceso, se abrirá la válvula V3 y se activará la bomba de vacío para extraer los gases acumulados en dicho depósito.
- Para la limpieza de la cámara de combustión de la máquina se realizará el mismo proceso que el de limpieza de depósito a diferencia que se activará de distinto botón y abrirá otra válvula V4.
- Tendrá que hacer una rutina de llenado de la cámara de combustión cuando el usuario pulse el botón virtual de llenado. El sistema gobernará tres válvulas, V1, V2 y V5 que dependerá su apertura y cierre de la presión en la que se encuentre en la cámara de combustión de la máquina. V1 se abrirá para el llenado rápido, V2 para el llenado lento. Estas dos válvulas darán el paso de gases de depósito a la cámara de combustión, mientras que V5 se abrirá cuando el llenado se quiera hacer con aire.
- El sistema tendrá que hacer un chequeo para verificar que la presión de llenado es la correcta. Este chequeo se realizará de forma automática una vez haya terminado el proceso de llenado.

4.3.3 Estructura de la programación

En este punto se va a explicar de forma estructurada las partes del programa que han sido realizadas para que cumpla los requisitos de las maniobras establecidas.

Gracias a los bloques de programación y la forma esquemática en la que se programa, se puede entender la mayor parte del software diseñado teniendo unos mínimos conocimientos de programación. A continuación se detallarán paso por paso cada maniobra del programa.

4.3.3.1 Estructura while loop

Esta estructura es una instrucción de Labview que hace que el programa se ejecute continuamente de forma cíclica mientras no sea pulsado el botón de paro de programa. Dentro de esta estructura se va a crear todo el programa necesario para el control de la máquina menos la configuración del puerto de comunicación y la de los pines de Arduino. En la siguiente figura 37 se muestra la estructura while loop sin programación en su interior, ya que no se puede mostrar todo el programa en pantalla.

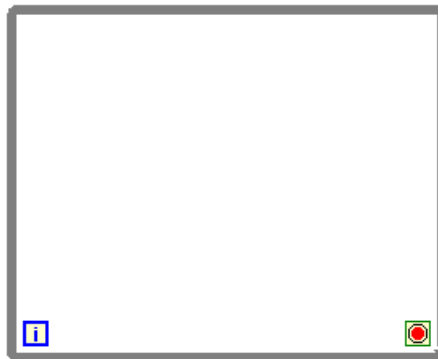


Figura 37 estructura while loop

4.3.3.2 Configuración de puerto de comunicación

En este punto se configura la comunicación que han de realizar entre el software y Arduino mediante la función Init.vi que viene dada de la librería LIFA para poder realizar la conexión. En esta configuración se tendrá que indicar varios parámetros como la velocidad de comunicación, tipo de puerto, modelo de Arduino y número de puerto. En la siguiente figura 38, se puede ver dicha configuración.

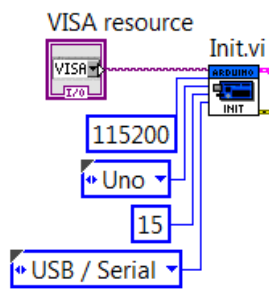


Figura 38 Configuración de puerto de comunicación

4.3.3.3 Configuración de pines de Arduino

En este apartado se va a ver como se configuran los pines de entradas y salidas de Arduino que van a ser utilizados para el control de la máquina. Los pines 10, 11, 12, 13 y 6 se han configurado como salidas para activar las respectivas electroválvulas V1, V2, V3, V4 y V5. El cableado de color rosa y amarillo que las une, es para la comunicación de los pines entre Labview y Arduino. En la siguiente figura 39 se muestra dicha configuración.

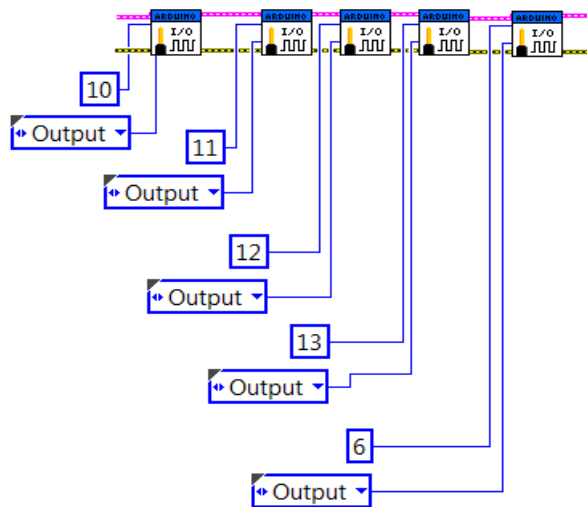


Figura 39 Configuración de pines de Arduino

4.3.3.4 **Indicador de limpieza de depósito**

Esta pequeña parte de programa indica mediante un piloto luminoso virtual que cuando la presión del depósito es menor o igual a la presión de consigna de limpieza del mismo, el proceso de limpieza ha terminado. A continuación, en la figura 40 podemos ver el fragmento de programa.

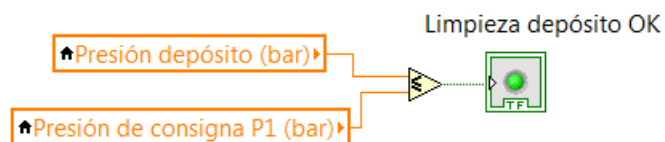


Figura 40 indicador de limpieza de depósito

4.3.3.5 **Indicador de limpieza de máquina**

Este indicador de limpieza se comporta igual que el de la limpieza de depósito pero comparando distinto sensor de presión y con distinta consigna. Ver en figura 41 el fragmento de programa.

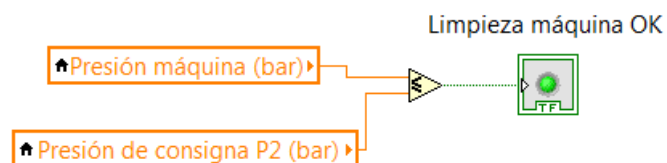


Figura 41 indicador de limpieza de máquina

4.3.3.6 **Indicador de llenado de máquina**

Este indicador de llenado avisa de que el proceso de llenado de la máquina ha sido completado. Este aviso se produce cuando la presión de la cámara de combustión es mayor o igual a la presión de consigna fijada por el usuario. En la figura 42 podemos ver el fragmento de programa.

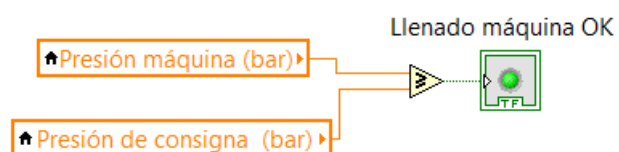


Figura 42 indicador de llenado de máquina

4.3.3.7 Activar y desactivar salidas de Arduino

En este punto vamos a ver como se tiene que configurar una salida de Arduino y quien da la orden para que en el pin de una determinada salida se obtenga el valor de 0 o 5Vdc. En la figura 43 que se va a ver a continuación se puede ver el fragmento de programa que hace dicha función.

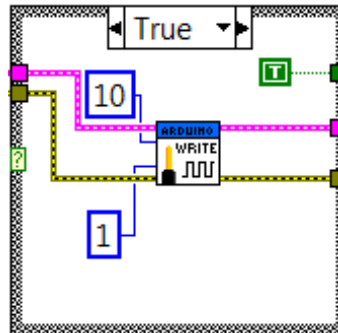


Figura 43 activar salida de Arduino

Como podemos ver en la figura 43, para activar una salida de Arduino, en este caso el pin 10, se ha utilizado una estructura case. Cuando activamos dicha estructura mediante una conexión en el selector case que es el interrogante verde que hay en el lateral derecho del cuadrado, pasa la estructura al estado true, es decir, se ejecuta la parte de programación que hay en su interior. En este caso lo que hace el programa es activar la salida del pin 10 poniéndole un 1 en la instrucción digital write. De esta forma se obtiene un valor de 5Vdc para activar el relé que va a dar paso a la alimentación de su electroválvula correspondiente, en este caso a V1, de llenado rápido.

Una vez se deshabilita la conexión de selector case, la estructura case cambia de estado y pasa a estar en false. Esto hace que se ejecute la parte de programación que está en su interior cuando está en ese estado. En la siguiente figura 44 se muestra el aspecto de la estructura case cuando cambia de estado.

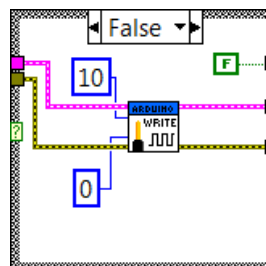


Figura 44 desactivar salida de Arduino

Como se puede observar en la figura 44, la estructura case ha pasado a estar al estado false. Cuando se cumple este evento, el programa desactiva la salida del pin 10 poniéndole un 0 en la instrucción digital write. De esta forma en la salida del pin 10 se obtendrá un valor de 0 Vdc que hará quitarle la alimentación a la electroválvula.

Esta parte de programación se ha repetido en los otros pines de salidas restantes como el pin 6, 11, 12 y 13 de Arduino, que corresponden a las respectivas electroválvulas V5, V2, V3 y V4. Las únicas diferencias que hay entre unas estructuras a otras, es que cada una se habilita de una señal distinta y que a cada una se le asigna el número de pin correspondiente a la instrucción digital write. En la siguiente figura 45 se muestra las estructuras case que activan las salidas 6, 10, 11, 12 y 13.

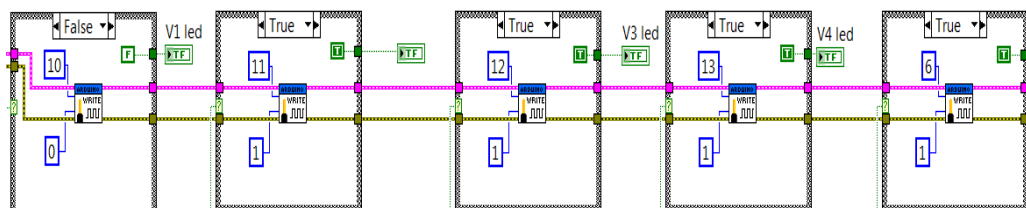


Figura 45 activar y desactivar salidas de Arduino

4.3.3.8 Lectura de presión de depósito

En este apartado se va a explicar cómo se hace la lectura de un sensor de presión mediante una entrada analógica de Arduino. Para una mejor comprensión se observará la siguiente figura 46 en la que se irán detallando los pasos seguidamente.

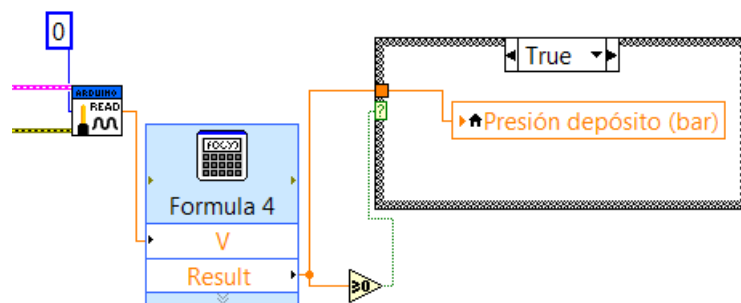


Figura 46 lectura de presión de depósito

Como se puede observar, este fragmento de programa está formado por tres partes. La primera está ubicada en la parte izquierda, esta es la configuración de la entrada analógica A0, se sabe que es la A0 por la constante de valor 0 que se le asigna a la instrucción analog read. Esta se encarga de transmitir al programa un valor en voltaje analógico que se encuentra en su pin de entrada.

La segunda parte es la fórmula, que se encuentra al medio del fragmento de programa. La función de esta instrucción es calcular mediante una fórmula creada por el programador, un valor que recibe en la entrada y seguidamente mostrar el resultado en la salida de la misma. En este caso, la instrucción fórmula, recibe el voltaje que ofrece el sensor de presión. Este voltaje está comprendido entre unos valores de 0,996 Vdc cuando la presión se encuentra a 0 bar y 4,98 Vdc cuando la presión está a 10 bar. Como el valor de la presión no es proporcional al voltaje que ofrece, se aplicará la fórmula de la regresión lineal. De esta forma, se consigue una linealidad en el muestreo de la presión. La fórmula es la siguiente.

$$\text{Presión (bar)} = 10 \times \frac{V_{in} - 0,996}{4,98 - 0,996}$$

La tercera y última parte es la estructura case, que se encuentra a la parte derecha del fragmento de programa. Cuando la presión sea mayor o igual que 0, la instrucción se pondrá en estado true y ejecutará lo que tenga en su interior programado. En este caso, se pondrá a escribir el valor de la presión del depósito en una variable local llamada presión depósito (bar). El valor de esta variable local será útil para leer su valor en otros puntos de programa que nombraremos más tarde.

4.3.3.9 *Lectura de presión de cámara de combustión*

La lectura de la presión en cámara de combustión se hace idénticamente que la lectura de presión del depósito que se ha visto en el punto anterior, ya que las características del sensor de presión son las mismas. Las únicas diferencias de dicha lectura es que el pin de entrada analógico es el A1, y que la variable local donde se escribe el valor de la presión se llama presión máquina (bar). En la figura 47 se muestra como se realiza la lectura de presión de la cámara de combustión.

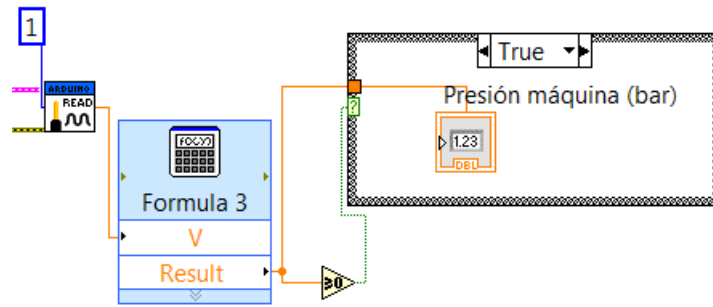


Figura 47 lectura de presión de máquina

4.3.3.10 Rutina de limpieza de depósito

En esta parte de programa se va a explicar cómo se realiza la rutina de limpieza de depósito. Para una mejor comprensión de esta parte de programación, se mostrará la siguiente figura 48 en la que se explicará con detalle las funciones programadas.

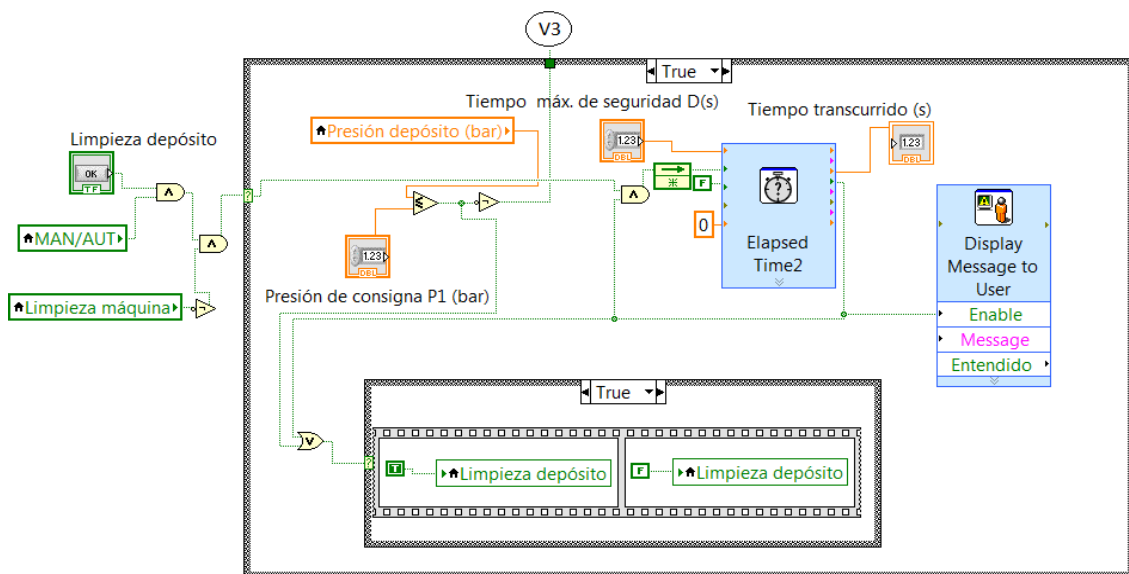


Figura 48 rutina de limpieza de depósito en estado true

Como se puede observar en la figura 48, la programación que controla la rutina de limpieza de depósito se encuentra en el interior de una estructura case. Esta estructura va a ser activada y pasará al estado true cuando se cumplan una serie de condiciones que se conectan en el punto de conexión selector case.

Para que dicha estructura pase al estado true, tendrá que estar el botón virtual de manual/automático en la posición de automático, el botón de limpieza depósito activado y por seguridad, no tendrá que estar el botón de limpieza de máquina activado.

Una vez se cumplan estas condiciones, el sistema se pone a comparar la presión del depósito con la presión de consigna de vaciado que se quiere alcanzar fijada por el usuario. Mientras la presión del depósito sea mayor a la presión de consigna, la electroválvula V3 y la bomba de vacío permanecerán activadas para evacuar los gases acumulados en el depósito.

Una vez se alcance la presión de consigna, dicha válvula y bomba serán desactivadas y la rutina de limpieza depósito se deshabilitará. Para que se desenclave el botón de limpieza de depósito y vuelva a la posición de reposo, se utilizará la estructura flat sequence. Dicha estructura se encuentra dentro de otra estructura case, y su función es cambiar de estado de forma secuencial el contenido de su interior. En este caso cambia de estado true (activado) a estado false (desactivado) la variable local de limpieza depósito que está vinculada con el botón. Para que esto ocurra, la estructura case que engloba a la estructura flat sequence, tendrá que activarse cuando termine el proceso de limpieza o cuando se supere el tiempo máximo de seguridad. Dicho tiempo de seguridad es programado para detectar algún posible fallo en la limpieza del depósito.

Unos de los posibles fallos que se pueden dar son las fugas de aire en la instalación, que no esté funcionando correctamente la electroválvula de limpieza, que la bomba de vacío no esté trabajando o que el sensor de presión esté dando una lectura errónea de la presión.

El tiempo de seguridad lo fijará el usuario desde el scada. Este tiempo tendrá que ser superior al tiempo de limpieza de depósito en condiciones ideales y sin ningún tipo de fallo en el sistema. De esta forma, si el tiempo de limpieza excede del fijado de seguridad, querrá decirse que el sistema no está funcionando correctamente.

Para crear la maniobra del control de tiempo de limpieza de depósito, se ha utilizado la instrucción elapsed time. Dicha instrucción se configura de la siguiente manera. Se introduce una constante de tiempo en el pin time target, que en este caso es la que fija el usuario como tiempo de seguridad, por otra parte, en el pin reset, se conecta la señal que da orden de inicio del sistema de la limpieza del depósito, también, hay que fijar una constante booleana false en el pin auto reset para hacer el reset de forma manual y por último, en el pin set start time se fijara una constante con valor 0 para que la cuenta de la instrucción elapsed time empiece por 0.

Una vez está la instrucción configurada y la rutina de limpieza activada, el sistema estará controlando el tiempo que dura la limpieza de depósito. Si el tiempo de limpieza supera al de seguridad fijado por el usuario, la instrucción dará una señal por el pin tim has elapsed.

Esta señal hará que pare la rutina de limpieza de depósito y mostrará un mensaje en pantalla avisando que el tiempo máximo de seguridad de limpieza

ha sido excedido. Una vez sea leído el mensaje, se pulsará sobre el botón de entendido y el mensaje desaparecerá.

Para poder mostrar dicho mensaje, se ha utilizado la instrucción display message to user. Dicha instrucción solo tendrá que recibir por el pin enable una señal para que muestre el mensaje que ha sido escrito por el programador.

Cuando una de las condiciones que activan la estructura case que engloba toda esta parte de programación que hemos nombrado en este punto no se cumple, la estructura pasará al estado false. En la figura 49 se puede ver el aspecto de la estructura case cuando paso al estado false.

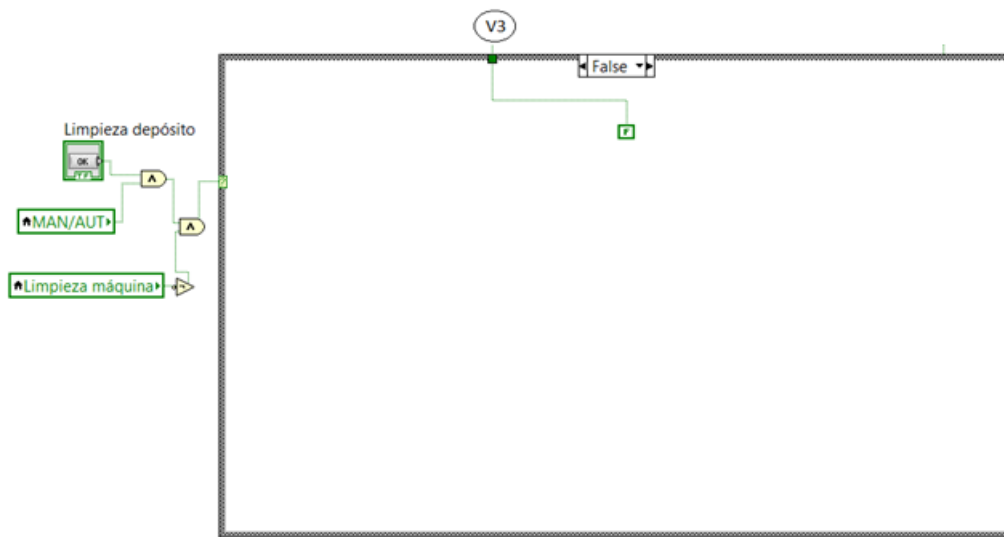


Figura 49 rutina de limpieza de depósito en estado false

Como se puede observar en la figura x, cuando pasa la estructura case al estado false, se desactiva la electroválvula V3 fijando un valor booleano false en la activación de su pin correspondiente.

En el apartado de explicación de estructura de scada se mostrarán todos los botones, indicadores de señales y variables numéricas que servirán de interface del usuario con el programa de control para el ajuste de esta rutina.

4.3.3.11 Rutina de limpieza de la cámara de combustión

Esta rutina es similar a la rutina de limpieza de depósito. Las únicas diferencias que hay de una rutina a otra son las siguientes: el botón de inicio de limpieza,

los indicadores de señales, las variables numéricas para introducir datos, el sensor de presión y la salida de la electroválvula. En la figura 50 se muestra la parte de programación que controla la rutina de limpieza de la cámara de combustión.

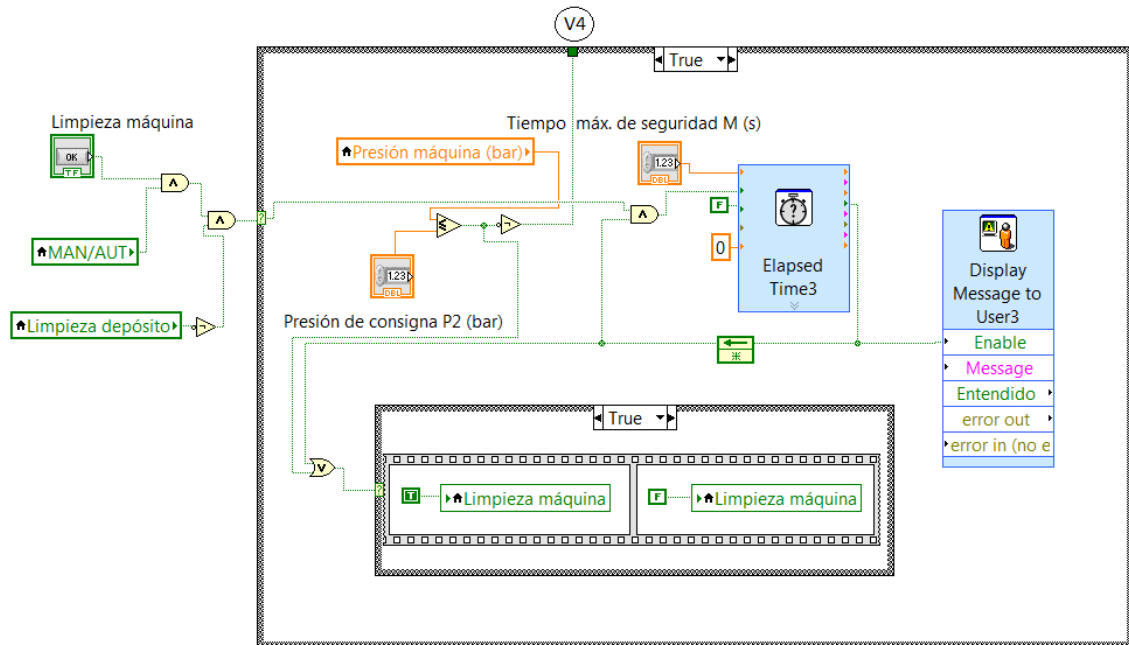


Figura 50 rutina de limpieza de cámara de combustión en estado true

En este punto no se va a entrar en detalles de los tipos y función que desarrollan cada una de las estructuras e instrucciones utilizadas en esta parte de programa, ya que son las mismas que las del punto anterior y ya han sido explicadas. Solo se va a explicar el funcionamiento de la rutina de limpieza de la cámara de combustión.

La rutina se iniciará cuando se cumplan las condiciones que activan la estructura case. Esta será activada cuando se cumplan las siguientes condiciones: esté pulsado el botón de limpieza máquina, esté el botón de manual/automático en la posición de automático y no esté activada la rutina de limpieza de depósito. Una vez se inicie la rutina de limpieza y la estructura case pase al estado true, el sistema comparará la presión de la cámara de combustión con la presión de consigna en la que se quiere hacer la limpieza. Mientras que la presión de la cámara de combustión sea mayor que la presión de consigna fijada por el usuario, la electroválvula V4 y la bomba de vacío permanecerán activas para extraer todos los gases residuales de las pruebas realizadas en la máquina. Una vez que la presión de la cámara sea menor o igual que la presión de consigna, la electroválvula V4 y la bomba de vacío se desactivarán, y la rutina de limpieza de cámara de combustión se deshabilitará e indicará con un piloto luminoso que el proceso de limpieza ha acabado.

Si el tiempo transcurrido desde el inicio de la rutina de limpieza, es superior al tiempo de seguridad fijado por el usuario, la electroválvula V4 se cerrará a la misma vez que la bomba de vacío se apagará de forma automática, seguidamente la rutina de limpieza se deshabilitará y mostrará un mensaje en pantalla indicando que ha sido excedido el tiempo máximo de limpieza. Este mensaje desaparecerá cuando el usuario confirme pulsando el botón confirmando que lo ha entendido. El mensaje será definido por el programador en una instrucción específica para este tipo de función.

Diagrama de la lógica de control para la limpieza de la máquina. Se muestran tres entradas: 'Limpieza máquina' (botón OK), 'MAN/AUT' (selección de modo) y 'Limpieza depósito' (botón de limpieza). Estas se combinan mediante puertas AND y OR para generar una señal que se conecta a un relé V4. El relé V4 está controlado por un interruptor 'False'.

Como se puede ver en la figura x, cuando la estructura case de la rutina de limpieza pasa al estado false, se desactiva la electroválvula V4 fijando un valor booleano false en la activación de su pin correspondiente.

En el apartado de explicación de estructura de scada se mostrarán todos los botones, indicadores de señales y variables numéricas que servirán de interface del usuario con el programa de control para el ajuste de esta rutina.

4.3.3.12 Rutina de llenado de la cámara de combustión

En este punto se va a explicar la parte de programación de la rutina de llenado de la cámara de combustión. Para entender mejor como ha sido estructurada la parte de programación de esta rutina, se va a mostrar la siguiente figura 52 donde se explicará seguidamente los elementos que la componen para que el sistema responda a las necesidades del usuario.

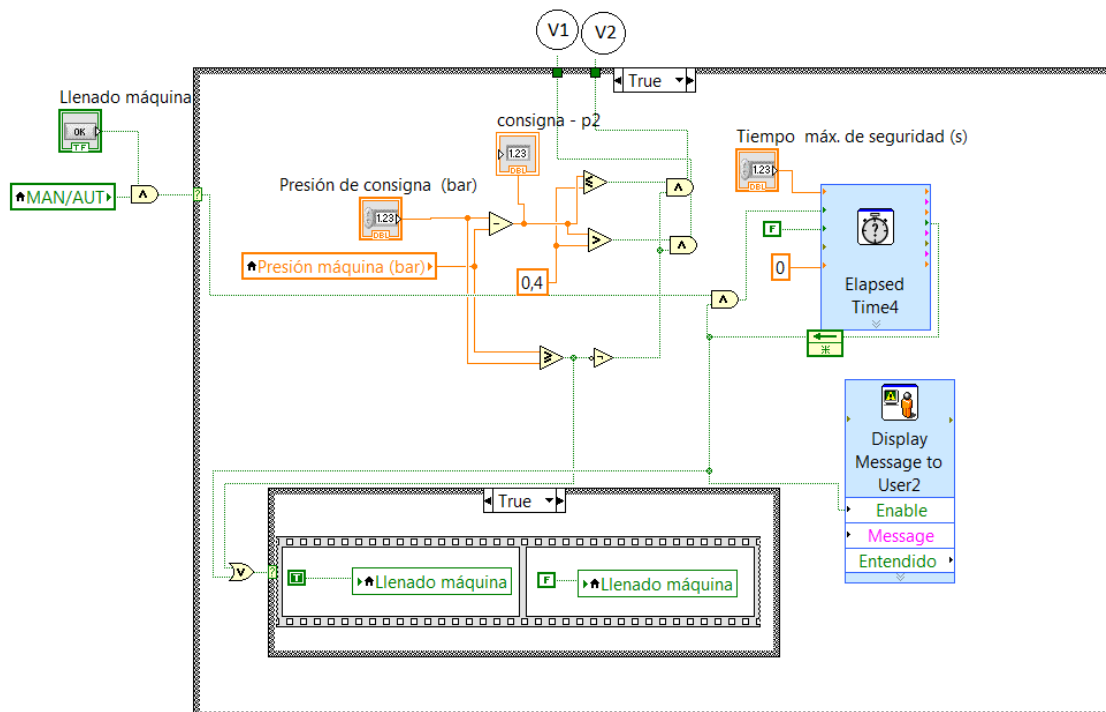


Figura 52 rutina de llenado de cámara de combustión en estado true

Como se puede observar en la figura x, la parte de programación que controla la rutina de llenado de la cámara de combustión, se encuentra dentro de una estructura case. Esta estructura se pondrá en estado true cuando las condiciones de inicio se cumplan. Las condiciones de inicio son las siguientes: el botón de llenado de la máquina tiene que estar pulsado y el botón de manual/automático tiene que estar en la posición de automático.

Una vez se cumplan las condiciones de inicio, la rutina de llenado se pondrá en marcha y la estructura case pasará al estado true. En este caso, el sistema se pondrá a comparar la presión de la cámara de combustión con la presión de

consigna de llenado fijada por el usuario desde el scada. Si la presión de la cámara de combustión, es menor a la presión de consigna fijada por el usuario, el sistema pondrá en marcha la rutina de llenado que constará de dos etapas. La primera etapa será la de llenado rápido, con un paso de caudal para la conducción de gases más amplio que la de llenado lento. Esta etapa mantendrá la electroválvula V1 activa mientras que haya una diferencia superior de 0,4 bar entre la presión de consigna y la presión de la cámara de combustión. Para obtener el resultado de la diferencia de presiones, se resta a la presión de consigna la presión que hay en la cámara de combustión. Este resultado se compara con una constante de valor 0,4, mientras que el resultado sea mayor de 0,4, el proceso de llenado se mantendrá en la primera etapa.

Una vez que la diferencia de presiones sea igual o menor al valor de 0,4, el sistema de llenado pasará a la segunda etapa de llenado. En esta etapa, la electroválvula V1 de llenado rápido se desactiva a la misma vez que se activa la electroválvula V2 de llenado lento. Dicha electroválvula se le llama de llenado lento porque el paso de caudal de gases es más estrecho que la de llenado rápido. Con este sistema se consigue que el llenado se realice de una forma rápida, y cuando se aproxime la presión de llenado a la presión de consigna termine el proceso de forma más precisa. Una vez se alcance la presión de consigna de llenado, la rutina de llenado se deshabilitará, desactivando la electroválvula V2 a la misma vez que activa la electroválvula V5 de aire ambiente, indicando con un piloto luminoso que el llenado está OK.

Como se puede apreciar en la figura x, Estos cálculos y comparaciones que se acaban de nombrar, se han realizado con instrucciones de comparación, numéricas, booleanas y variables locales.

Para garantizar que la presión de llenado de la cámara de combustión se mantiene al mismo valor que la presión de consigna deseada, se realiza un chequeo después de unos segundos de haber terminado la rutina llenado. De esta forma, se asegura de que no ha habido ninguna pérdida de presión en el llenado para realizar la prueba de combustión de forma correcta.

Para realizar la maniobra de chequeo se ha necesitado la señal de presión de consigna alcanzada. Cuando se da esta señal, activa un temporizador de estructura elapsed time, que después de 2 segundos vuelve a chequear si la presión de la cámara de combustión es igual o mayor a la presión de consigna. Si la presión es la correcta, la rutina de llenado se deshabilita, y si la presión es incorrecta, el sistema abre otra vez las electroválvulas para completar el llenado. Este proceso se repetirá hasta que la presión de llenado se mantenga estable.

Otra parte que hay que destacar, es que, siempre que V1 o V2 estén activas mientras esté en marcha la rutina de llenado, la electroválvula V5 tiene que permanecer desactivada. De esta forma nos aseguramos que los gases son

enviados a la cámara de combustión y no al aire ambiente. Por otra parte, se debe cumplir que siempre que este el sistema en modo automático con V1 y V2 desactivadas, debe permanecer V5 activada por si se quiere realizar un llenado con aire. Para entender mejor la maniobra que gobierna la electroválvula V5, se mostrará la siguiente figura 53

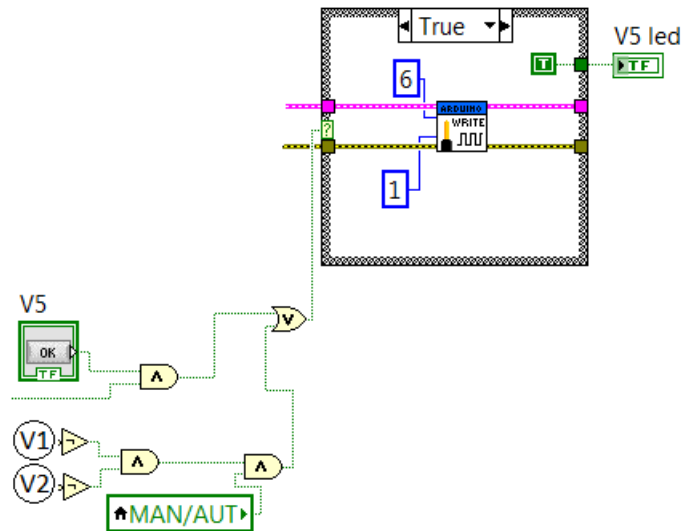


Figura 53 activación, desactivación de electroválvula V5 en estado true

En la figura 53 se puede ver la combinación de instrucciones booleanas que han sido necesarias para la activación de la electroválvula V5. Como bien indica esta maniobra, siempre que estén V1 y V2 desactivadas, y el proceso en automático, la estructura case pasará al estado true. Cuando esté en este estado, se pondrá un 1 en la en el pin 6 de Arduino para activar la electroválvula V5.

Cuando no se cumplan las condiciones que se han nombrado anteriormente, la estructura case pasara al estado false, desactivando a la misma vez la electroválvula V5. Ver en la siguiente figura 54 el aspecto de la estructura case cuando está en estado false.

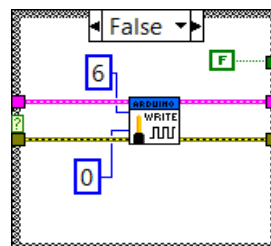


Figura 54 activación, desactivación de electroválvula V5 en estado false

Como bien muestra la figura x, cuando las estructura case pasa al estado false, se pone un 0 en el pin 6 de Arduino. De esta forma se desactiva la salida que alimenta a la electroválvula V5 de aire ambiente.

Volviendo a la rutina de llenado, hay que indicar que también dispone de un tiempo máximo de seguridad de llenado. Este tiempo será fijado por el usuario en el scada del programa. El tiempo fijado tendrá que ser superior al tiempo de funcionamiento en condiciones ideales y con la ausencia de algún tipo de avería en el sistema. Una vez que la rutina de llenado se inicie, se empezará a contar el tiempo transcurrido mediante la instrucción de control de tiempo elapsed time. Cuando se alcance el tiempo máximo de seguridad, las electroválvulas V1 y V2 se cerrarán y la rutina de llenado se deshabilitará, mostrando a la misma vez un mensaje en pantalla diciendo que ha sido excedido el tiempo máximo de llenado. El mensaje será escrito por el programador, para que se muestre en pantalla con la ayuda de la instrucción display message.

Una vez que no se cumplan la condiciones que habilitan la estructura case de rutina de llenado, dicha estructura pasará al estado false. En la siguiente figura 55 se puede ver la parte de programación del interior de la estructura case cuando esta se encuentra en estado false.

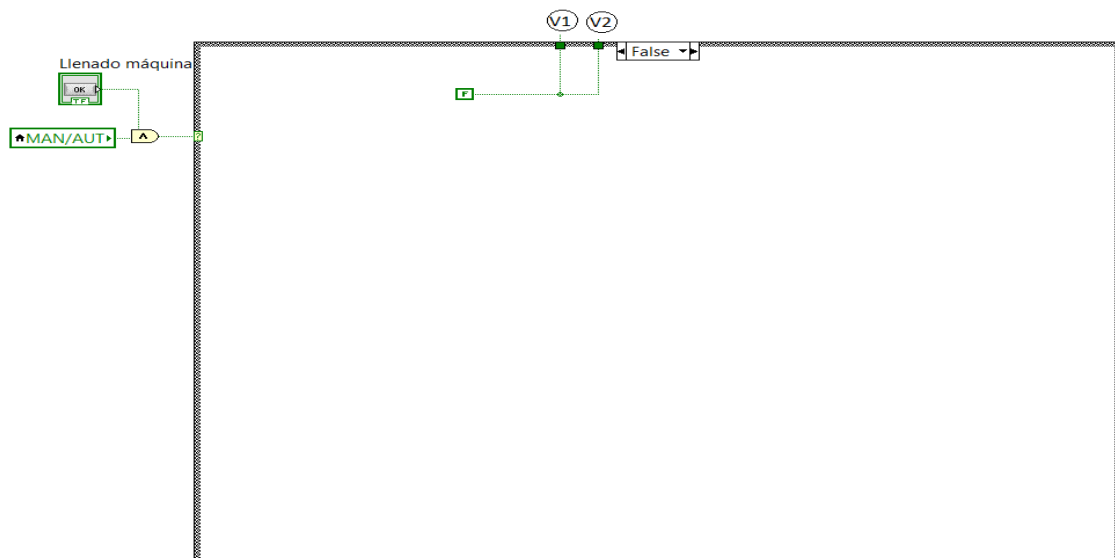


Figura 55 rutina de llenado de cámara de combustión en estado false

En la figura 55 se muestra la parte de programación que hay en el interior de la estructura case de la rutina de llenado cuando pasa al estado false. Cuando se encuentra en este estado, se desactivan las electroválvulas V1 y V2 fijándole una señal false tipo booleana en la activación de sus pines de salida correspondientes.

En el siguiente punto 4.3.4 de estructura de scada, se mostrarán todos los botones, indicadores de señales y variables numéricas que servirán de interface del usuario con el programa de control para el ajuste de esta rutina.

4.3.4 Estructura del scada

En este apartado, se va a mostrar y explicar toda las partes que forman el scada para controlar el sistema de llenado y limpieza de la máquina de compresión rápida. Este interface también ha sido creado con Labview, que permite vincular parámetros de lectura y escritura con el programa creado con el mismo software.

Para familiarizarse un poco con el scada, se va a mostrar una vista general en la siguiente figura 56, seguidamente se explicará el funcionamiento de cada una de sus partes.



Figura 56 vista general del scada

Como se puede observar en la figura 56, el scada está estructurado en varios bloques. Cada uno de ellos tiene sus elementos para la lectura o escritura de datos que se procesan en su fragmento de programa asignado. Dicho interface está formado por 6 bloques y dos botones principales. Los bloques que lo forman son los siguientes: empezando por arriba y por la izquierda, se sitúa el bloque de limpieza de depósito, justamente debajo, se encuentra el bloque de limpieza de máquina, debajo de este, está el llenado de máquina, y debajo del bloque de llenado se encuentra el bloque de los botones de forzado en manual y los indicadores luminoso de las electroválvulas que pertenecen al sistema.

Empezando por arriba a la derecha, se encuentra el indicador de presión instantánea del depósito, justamente debajo, está situado el indicador de

presión instantánea de la máquina, y por último, debajo de este, se sitúa el bloque de comunicación de Arduino con Labview. Los botones principales son el de parar el programa que está situado a la derecha, y el manual/automático que está situado a la izquierda del botón de parada. Como se ha comentado anteriormente, se va a explicar en los puntos siguientes los bloques que forman el scada de forma individual.

4.3.4.1 *Bloque de limpieza depósito*

Esta parte del scada, se encarga de intercambiar datos con el usuario para llevar el ajuste e inicio del control de la rutina de limpieza de depósito. Para una mejor comprensión de la explicación, se va a mostrar la siguiente figura 57 que contiene el bloque que va a ser explicado.



Figura 57 proceso de limpieza de depósito activado

Dentro del bloque de limpieza de depósito, está disponible una serie de parámetros para que el usuario haga el ajuste necesario dependiendo sus necesidades y las de la máquina. Uno de ellos es la introducción de la presión de consigna. Esta presión se fija al valor en el que se quiere alcanzar la presión de limpieza. El otro parámetro de ajuste es el de tiempo máximo de seguridad, que fijara el usuario con un tiempo mayor de la rutina de limpieza en condiciones ideales y sin ningún tipo de avería en el sistema.

También se dispone de un interruptor de marcha de limpieza depósito. Cuando este es pulsado, cambia de un color gris a un color verde. Esto indica que la

rutina de limpieza está en proceso, y cuando pasa de color verde a color gris, indica que la rutina de limpieza ha terminado o que ha habido un fallo en el sistema por alcanzar el tiempo máximo de seguridad.

Si se diera el caso de que se alcanza el tiempo máximo de seguridad, a la misma vez que deshabilita la rutina de limpieza, muestra un mensaje de alarma como el que se va a mostrar en la siguiente figura 58.

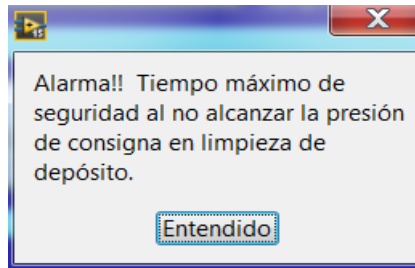


Figura 58 mensaje de alarma de limpieza depósito

Una vez se lea el mensaje, se pulsará en el botón de entendido y este desaparecerá y el sistema se pondrá en reposo.

Por otra parte también cuenta con un piloto luminoso que indica que la limpieza del sistema está en OK, es decir, que la limpieza esta completada. Como se puede ver en la figura x, el botón de limpieza de depósito está en verde, que como bien se ha indicado anteriormente, hace referencia a que la rutina de limpieza está en marcha. Mientras el botón permanezca en verde, el piloto luminoso de la electroválvula V3 también permanecerá en verde, hasta que la presión del depósito sea menor que la de consigna o se alcance el tiempo máximo de seguridad.

En la siguiente figura 59, se va a mostrar el aspecto que tiene el scada cuando la limpieza de depósito ha sido completada.



Figura 59 proceso de limpieza de depósito completado

Si se observa la figura 59, se puede ver que está en verde el piloto que indica que la limpieza de depósito está en OK. Esto indica que la presión del depósito está por debajo de la presión de consigna. En el scada se puede verificar mirando las presiones que la rutina de limpieza ha sido completada. Cuando esto se cumpla, el piloto de la electroválvula V3 se apagará, y el botón de inicio llenado se deshabilitará pasando de un color verde a un color gris.

4.3.4.2 *Bloque de limpieza máquina*

Este bloque es muy similar al de limpieza de depósito que ha sido explicado en el punto anterior. La única diferencia es que cada rutina controla la limpieza una parte distinta del sistema.

Para ver mejor el comportamiento de los elementos que forman parte del bloque de limpieza de máquina, se va a mostrar la siguiente figura 60.



Figura 60 proceso de limpieza de máquina activado

Como se acabada de comentar, la limpieza de máquina tiene los mismos ajustes que el bloque de limpieza de depósito. Ajuste del valor de presión consigna y ajuste del tiempo máximo de seguridad. También dispone del botón de marcha de limpieza de máquina y del indicador luminoso de limpieza máquina OK.

En la figura 60 se puede observar que el botón de limpieza máquina está de color verde, esto indica que la rutina de limpieza se encuentra activa. Este botón junto con el indicador luminoso de la electroválvula V4, permanecerán activos hasta que se alcance la presión de consigna de limpieza de máquina o la rutina se pare por el tiempo máximo de seguridad.

Si se diera el caso de que se alcanzara el tiempo máximo de seguridad, se deshabilitaría la rutina de limpieza máquina a la misma vez que mostraría un mensaje en pantalla avisando de que se ha alcanzado dicho tiempo. En la siguiente figura 61 se puede ver el aviso que se muestra en pantalla.

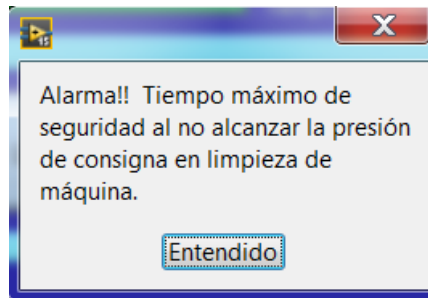


Figura 61 mensaje de alarma de limpieza máquina

Una vez se haya leído el mensaje y se quiera continuar trabajando con el scada, se pulsará el botón de entendido y desaparecerá el mensaje dejando el sistema listo para seguir trabajando.

Cuando se alcance la presión de consigna de limpieza de la máquina, la rutina de limpieza se deshabilitará y el scada cambiará de estado como se va a mostrar en la siguiente figura 62.



Figura 62 proceso de limpieza de máquina completado

Una vez se alcance la presión de consigna de la limpieza de la máquina, se iluminará el piloto de llenado máquina OK. La rutina de limpieza se deshabilitará, pasando al estado de reposo el piloto luminoso de la electroválvula V4 y el botón de limpieza máquina. Como bien se puede observar en el indicador de presión máquina, la presión está por debajo de la presión de consigna.

4.3.4.3 *Bloque de llenado de máquina*

Este bloque se encarga de ofrecer el control y el ajuste de parámetros necesarios al usuario para llevar a cabo la rutina de llenado de la máquina. toda esta parte de control es similar a las rutinas de limpieza que se han nombrado anteriormente. Para tener una mejor idea del comportamiento del scada cuando se habilita la rutina de llenado, se verá la siguiente figura 63.



Figura 63 proceso de llenado de máquina etapa 1 activado

Cuando se habilite la rutina de llenado de la máquina, se desactivará la electroválvula V5 y se activará la electroválvula V1 mientras que la diferencia de presión entre la consigna de llenado y la presión en la cámara de combustión sea mayor de 0,4bar. Cuando dicha diferencia de presiones sea menor de 0,4bar, se desactivará la electroválvula V1 de llenado rápido y se activará la electroválvula V2 de llenado lento. Si se observa en la siguiente figura 64, se aprecia cómo se ha activado la electroválvula V2 al haber una diferencia menor de 0,4 bar entre la presión de consigna de llenado y la presión de la cámara de combustión.



Figura 64 proceso de llenado de máquina etapa 2 activado

En la figura 64 se puede ver que las diferencias de presiones entre la consigna de llenado de 4 bar, y la cámara de combustión de 3,67 bar, es menor de 0,4 bar. Mientras que esta diferencia esté entre el rango de 0,4 a 0,1 bar, la electroválvula V2 permanecerá activa para completar el llenado de forma precisa.

Una vez se alcance una diferencia de 0 bar, o haya excedido el tiempo máximo de seguridad, la rutina de llenado se deshabilitará y el sistema se pondrá en reposo.

Si se diera el caso de que el tiempo de ejecución de la rutina de llenado de máquina alcanzase el tiempo máximo de seguridad, la rutina de llenado se deshabilitaría como se acaba de comentar y mostraría el mensaje de alarma de la siguiente figura 65.

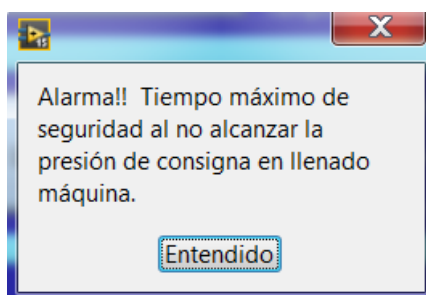


Figura 65 mensaje de alarma de llenado de máquina

Cuando se complete el llenado de la máquina, el scada tendrá la apariencia de la siguiente figura 66.



Figura 66 proceso de llenado de máquina completado

Esta figura muestra como la presión de la máquina es mayor a la de consigna, esto apunta que se ha completado el llenado como bien indica el piloto luminoso de llenado máquina OK. Cuando se alcanza este estado, el botón de llenado de máquina y el piloto luminoso de la electroválvula V2, se desactivan a la misma vez que se activa el piloto luminoso de la electroválvula V5.

4.3.4.4 **Bloque de indicadores y forzados manuales de electroválvulas**

Este bloque se ha creado para realizar dos funciones. Una de ellas es indicar al usuario mediante pilotos luminosos el estado en que se encuentra cada electroválvula. La otra función es dar la opción al usuario de poder forzar todas las electroválvulas de forma individual desde su botón correspondiente mientras esté el sistema en manual. En la siguiente figura 67 se puede ver el forzado de todas las electroválvulas mientras el sistema está de forma manual.



Figura 67 forzado de las electroválvulas en manual

4.3.4.5 **Bloque de muestreo de presión en depósito**

Este bloque tiene como función muestrear la presión instantánea para que el usuario pueda ver la presión en la que se encuentra el depósito. El bloque de muestreo de presión de depósito se puede ver en la siguiente figura 68.

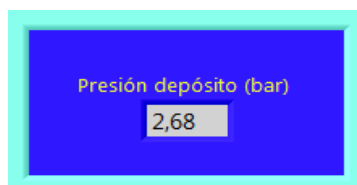


Figura 68 presión instantánea de depósito

4.3.4.6 **Bloque de muestreo de presión en máquina**

Este bloque igual que el bloque de muestreo de presión de depósito, tiene la misma función, pero en este caso para ver la presión instantánea en la que se encuentra la cámara de combustión de la máquina. El bloque de muestreo de presión de máquina se puede ver en la siguiente figura 69.

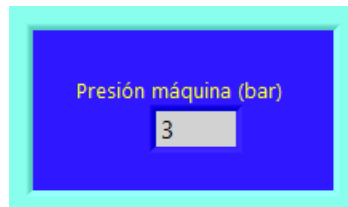


Figura 69 presión instantánea de máquina

4.3.4.7 **Bloque de comunicación de Arduino con Labview**

Este bloque ha sido creado exclusivamente para llevar un control de la comunicación entre Arduino y Labview. Viendo la siguiente figura 70 y la explicación posterior, se podrá tener una idea de su funcionalidad.

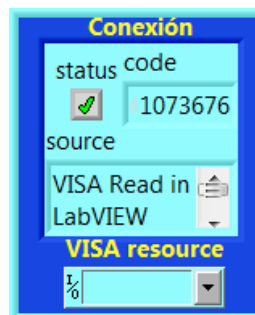


Figura 70 bloque de comunicación Arduino y Labview en buen estado

En él se puede observar si hay algún tipo de error en la comunicación, mostrando un código que indica el tipo de fallo que se ha provocado. También dispone del indicador status, que informa de si el sistema está funcionando correctamente. En otra ventana describe la función que está realizando la comunicación, en este caso el software Visa está leyendo el interface de Labview con Arduino. Por último en la parte inferior del bloque de comunicación, se introduce el número de puerto en el que está conectado el ordenador con la placa Arduino. En este caso, el número de puerto que se está utilizando para la aplicación es el COM3. Si el sistema no consigue que comunique Arduino con Labview, el bloque de comunicación tendrá una apariencia como la siguiente figura 71.

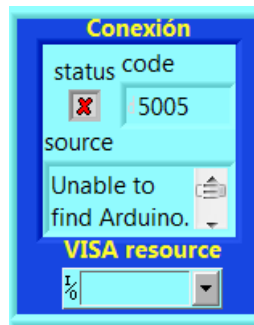


Figura 71 bloque de comunicación Arduino y Labview en mal estado

Viendo la figura x se puede observar que ha cambiado la apariencia del bloque de comunicación. Este aspecto lo muestra cuando no se puede establecer una comunicación entre Arduino y Labview. Como se puede ver en dicha figura, el estatus ahora muestra una X roja, el código de error es el 5005 y el mensaje que muestra, informa de que es incapaz de encontrar Arduino.

4.3.4.8 **Botón de manual/automático**

Este botón se ha implementado en el scada para poder poner el sistema de forma manual o de forma automática. Cuando está en la posición de manual, el sistema nos permite hacer un forzado de todas las electroválvulas una vez se pulse el botón que corresponde a cada una de ellas. Por otra parte restringe el funcionamiento de las rutinas de limpieza y de llenado. En la siguiente figura 72 se puede ver el botón manual/automático cuando está en la posición de manual.



Figura 72 botón manual/automático en posición manual

Cuando dicho botón es pulsado, cambia su estado a automático. Si se encuentra el botón en automático, permite la marcha de las rutinas de limpieza y llenado a la misma vez que bloquea el forzado de las electroválvulas. Este tipo de bloqueo se ha implementado para que no se pueda forzar ninguna electroválvula mientras se esté ejecutando alguna rutina. De esta forma se evitarán posibles confusiones y accidentes en las pruebas realizadas por el usuario. En la siguiente figura 73 se muestra el aspecto del botón cuando cambia a la posición de automático.



Figura 73 botón manual/automático en posición automático

4.3.4.9 Botón de paro general del sistema

Este botón lo crea el software de Labview de forma automática cuando se utiliza la estructura while loop en el diagrama de bloques. Dicho botón se utiliza exclusivamente para detener el programa en cualquier momento que el usuario desee. Una vez se pulsa, se para por completo toda la parte de programación que se encuentra en el interior de la estructura while loop. En la siguiente figura 74 se muestra el botón de paro general del sistema.



Figura 74 botón de paro general del sistema

5. Comprobación del sistema

5.1 Introducción

En este apartado se va a explicar el proceso que se ha realizado para la comprobación del funcionamiento del sistema. Este proceso es necesario para confirmar que el funcionamiento lo hace de forma correcta. Para ello, se han realizado las pruebas en la máquina donde se ha implementado el proyecto que se ha llevado a cabo. En los siguientes puntos se explicará con detalle las pruebas que se han efectuado.

5.2 Pruebas realizadas

5.2.1 Comprobación del cableado de la instalación

Antes de darle tensión al sistema, se ha comprobado que todo el cableado de la instalación está conectado de forma correcta. Para confirmar que es así, se ha utilizado un polímetro con el que se ha medido la resistencia en los puntos cableados.

Primeramente se ha medido resistencia entre las entradas y salidas de la placa Arduino, y los pines del conector que está en la caja del circuito impreso. Seguidamente se ha medido resistencia en los extremos del cable conector que une la caja del circuito impreso con la caja de conexiones situada en la máquina. Por último se ha comprobado la resistencia entre los pines del conector de la caja de conexiones de la máquina, y las electroválvulas y sensores que hay repartidos en la misma máquina.

Este proceso se ha realizado para confirmar que cada entrada o salida de Arduino corresponde con su electroválvula o sensor.

5.2.2 Comprobación de las tensiones de alimentación

Una vez se ha comprobado que todas las conexiones se han hecho de forma correcta, se le suministra la alimentación al sistema y se comprueba con el polímetro que las tensiones son correctas. Primeramente se alimenta la fuente de alimentación a una tensión de 220Vac. Una vez se ha alimentado, se comprueba que a la salida de la misma fuente de alimentación haya un valor de 24Vdc. Seguidamente se ha comprobado que a la salida del estabilizador de tensión que ha sido alimentado a 24Vdc, tenga un valor de 12Vdc para alimentar la placa Arduino. Por último, se ha comprobado que la alimentación

del común del contacto de lo relés que controlan las electroválvulas tenga una tensión de 24Vdc.

5.2.3 Comprobación de la comunicación entre Arduino y Labview

Cuando ya se ha alimentado el sistema, se comprueba que haya comunicación entre Arduino y Labview. Antes de realizar la comunicación, hay que asegurarse con que puerto serie se va a realizar la comunicación. Para saber en qué puerto está conectada la placa Arduino al ordenador, hay que buscar en el menú de Windows en panel de control, y dentro de panel de control, el menú de administrador de dispositivos. En la siguiente figura 75 se muestra la ruta que indica el puerto serie utilizado.

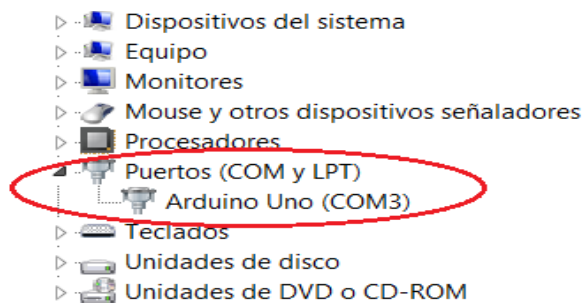


Figura 75 número de puerto serie de comunicación

Como bien muestra la figura 75, se puede ver que dentro del menú de administrador de dispositivos se encuentra el desplegable de puertos COM y LPT. Si se pulsa dicho desplegable, se muestran todos los dispositivos conectados al ordenador con su número de puerto. En este caso solo tenemos uno, y es el puerto COM 3 que está siendo utilizado por la placa Arduino.

Una vez se obtiene el número de puerto de comunicación, se pone el programa en marcha del sistema y se introduce en el bloque de comunicación del scada. En la siguiente figura 76 indica donde se introduce el número de puerto.

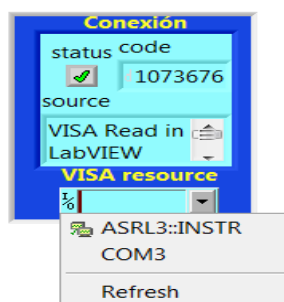


Figura 76 introducción del número de puerto de comunicación

Cuando ya se ha configurado el número de puerto y está el ordenador conectado a la placa Arduino con el conector de puerto USB, se da marcha al sistema y se comprueba de forma visual en el scada que no hay ningún tipo de fallo.

Si se siguen esta serie de pasos el sistema debe funcionar correctamente.

5.2.4 Comprobación de las entradas y salidas de Arduino

Una vez se ha comprobado que el cableado de la instalación y la comunicación del puerto está correctamente, se prueban las entradas y las salidas de Arduino. Para comprobar que las electroválvulas responden a las órdenes del sistema, se pone el sistema en manual y se van forzando de una en una las electroválvulas con su botón correspondiente del scada. Si se fuerzan y las electroválvulas cambian de estado, se confirma que las salidas de Arduino están funcionando de forma correcta.

Por otra parte, se hace la comprobación de las entradas analógicas. Para poder hacer esta comprobación hay que variar la presión del conducto donde estén ubicadas. La prueba se puede hacer introduciendo algún tipo de gas. Una vez se introduzca el gas, hay que observar en el scada que la presión está variando, si es así, los sensores de presión están respondiendo a dichas variaciones, por lo cual se confirma que también están funcionando.

5.2.5 Comprobación de las rutinas de limpieza y llenado

Cuando ya se tiene el sistema listo en todos los aspectos, se procede a la comprobación de las rutinas de forma automática. Antes de dar marcha a cualquier rutina, se introducen los parámetros de ajuste que se desea en cada bloque de proceso de limpieza y de llenado.

Una vez están todos los parámetros introducidos, se pone el sistema en modo automático y se pulsa la rutina que se quiera probar. Cuando está la rutina en marcha, hay que comprobar que responde el sistema al cierre y apertura de las electroválvulas dependiendo de las presiones fijadas de consigna. Por otra parte, también hay que hacer la comprobación de parada del sistema cuando no se ha alcanzado la presión en un tiempo máximo de seguridad.

Si el resultado de las pruebas es el correcto, el sistema está listo para trabajar.

6. Conclusiones y posibles mejoras

6.1 Conclusiones

La conclusión que se ha obtenido en este proyecto, es que con un bajo coste en materiales y unos conocimientos básicos en programación y electrónica, se pueden hacer proyectos muy interesantes para realizar controles automáticos. En mi opinión, si se quiere controlar un proceso industrial con muchas más entradas y salidas, e instrucciones más complejas, utilizaría autómatas industriales. El único inconveniente que hay es el coste, que es mucho más elevado que los materiales empleados en este proyecto. Por otra parte, quiero aclarar, que el lenguaje de programación de los autómatas me agrada más que el lenguaje C++ que se utiliza con el software Labview.

6.2 Posibles mejoras

Posibles mejoras no se han planteado por el momento, ya que el sistema responde correctamente a los requisitos fijados para el control del sistema. Tampoco se descarta que con el tiempo se agregue alguna maniobra más si se decide implementar algún otro tipo de control en la máquina de compresión rápida.

7. Presupuesto

7.1 Introducción

En este punto se van a detallar los costes de materiales y de mano de obra que han sido necesarios para poder realizar el proyecto. Estos gastos van a ser desglosados en los siguientes puntos que se nombraran a continuación. Hay que destacar, que también se han utilizado recursos informáticos para desarrollar el proyecto y no se han incluido en el presupuesto, ya que se disponía previamente de ellos. Los recursos informáticos utilizados son los siguientes: Licencia Labview 2015, Licencia IDE Arduino, Software LIFA de Labview, Licencia Microsoft Office 2010, Ordenador de sobre mesa del laboratorio y Ordenador Portátil ASUS x54c.

7.2 Gastos en materiales

En este apartado se van a desglosar todas las cantidades y precios de los materiales que han sido utilizados en el proyecto. En la siguiente tabla se muestra de forma estructurada los gastos generados.

Materiales	unidades	coste unidad (€)	importe(€)
Condensador 470µF	1	0,12	0,12
Condensador 2200µF	1	0,71	0,71
Placa circuito impreso	1	13,57	13,57
Bornes de conexión	8	0,37	2,96
Regulador de tensión	1	0,38	0,38
Amplificador ULN2003AN	1	0,6	0,6
Relé RE030024	6	2,8	16,8
Resistencia 249Ω	2	1,51	3,02
Fuente de alimentación	1	48,91	48,91
Caja de conexiones	2	15,23	30,46
conector MIDI	1	7,82	7,82
Manguera de 12 hilos	1	12	12
Arduino UNO	1	23,5	23,5
Electroválvula SITE	5	69,37	346,85
Sensor de presión WIKA	2	85,07	170,14
TOTAL (Iva incluido)			677,84

7.3 Gastos de mano de obra

En este punto se va a desglosar el coste de las horas empleadas del personal cualificado para el desarrollo del proyecto. Los costes de las horas que se han contabilizado son arreglo al convenio correspondiente. En la siguiente tabla se puede ver los gastos generados por el personal que ha intervenido en el proyecto.

Personal	Horas	coste unidad (€/h)	importe(€)
Profesor tutor	30	23,33	699,9
Ingeniero industrial	100	11,67	1167
Técnicos	50	10	500
Alumno (Ingeniero)	350	11,67	4084,5
TOTAL (Iva incluido)			7806,194

7.4 Gastos totales

Como bien se ha detallado en el punto 7.3, los gastos totales en material, ascienden a una cantidad de 677,84€, y los gastos en mano de obra obtenidos en el punto 7.4 ascienden a un total de 7806,19€. La suma total de los gastos generados para poder desarrollar el proyecto, asciende a un importe de 8484.03€.

8. Bibliografía

El siguiente listado hace referencia a toda la herramienta teórica que ha sido necesaria para el desarrollo del proyecto que se ha llevado acabo.

- (1) Foros Arduino. <https://forum.arduino.cc/>
- (2) Foros de national instruments. <http://forums.ni.com/>
- (3) Boylestad, Nashelsky (2009). Teoría de circuitos y dispositivos electrónicos
- (4) Ruíz J.M (2012). Herramientas gráficas para la programación de Arduino